Aix-Marseille Université
Université de Sherbrooke
École doctorale 184
UFR Sciences
Laboratoire des Sciences de l'Information et des Systèmes (LSIS)/G-Mod
Centre d'Applications et de Recherches en Télédétection (Cartel)

Thèse présentée pour obtenir le grade universitaire de docteur

Discipline : Mathématiques et Informatique
Spécialité : Informatique

Joris RAVAGLIA

Reconstruction de formes tubulaires à partir de nuages de points : application à l'estimation de la géométrie forestière

Soutenue le 14 décembre 2017 devant le jury :

Raphaël PÉLISSIER	UMR AMAP	Rapporteur
Franck HETROY	ICube, Université de Strasbourg	Rapporteur
Géraldine MORIN	IRIT, Université de Toulouse	Rapporteur
Éric GALIN	LIRIS, Université Lyon 1	Examinateur
Richard Fournier	CARTEL, Université de Sherbrooke	Directeur de thèse
Alexandra Bac	LSIS, Aix-Marseille Université	Directeur de thèse
Marc DANIEL	LSIS, Aix-Marseille Université	Directeur de thèse

Résumé

Les capacités des technologies de télédétection ont augmenté exponentiellement au cours des dernières années : de nouveaux scanners fournissent maintenant une représentation géométrique de leur environnement sous la forme de nuage de points avec une précision jusqu'ici inégalée. Le traitement de nuages de points est donc devenu une discipline à part entière avec ses problématiques propres et de nombreux défis à relever. Le cœur de cette thèse porte sur la modélisation géométrique et introduit une méthode robuste d'extraction de formes tubulaires à partir de nuages de points. Nous avons choisi de tester nos méthodes dans le contexte applicatif difficile de la foresterie pour mettre en valeur la robustesse de nos algorithmes et leur application à des données volumineuses.

Nos méthodes intègrent les normales aux points comme information supplémentaire pour atteindre les objectifs de performance nécessaire au traitement de nuages de points volumineux. Cependant, ces normales ne sont généralement pas fournies par les capteurs, il est donc nécessaire de les pré-calculer. Pour préserver la rapidité d'exécution, notre premier développement a donc consisté à présenter une méthode rapide d'estimation de normales. Pour ce faire nous avons approximé localement la géométrie du nuage de points en utilisant des "patchs" lisses dont la taille s'adapte à la complexité locale des nuages de points.

Nos travaux se sont ensuite concentrés sur l'extraction robuste de formes tubulaires dans des nuages de points denses, occlus, bruités et de densité inhomogène. Dans cette optique, nous avons développé une variante de la transformée de Hough dont la complexité est réduite grâce aux normales calculées. Nous avons ensuite couplé ces travaux à une proposition de contours actifs indépendants de leur paramétrisation. Cette combinaison assure la cohérence interne des formes reconstruites et s'affranchit ainsi des problèmes liés à l'occlusion, au bruit et aux variations de densité. Notre méthode a été validée en environnement complexe forestier pour reconstruire des troncs d'arbre afin d'en relever les qualités par comparaison à des méthodes existantes.

La reconstruction de troncs d'arbre ouvre d'autres questions à mi-chemin entre foresterie et géométrie. La segmentation des arbres d'une placette forestière est l'une d'entre elles. C'est pourquoi nous proposons également une méthode de segmentation conçue pour contourner les défauts des nuages de points forestiers et isoler les différents objets d'un jeu de données.

Durant nos travaux nous avons utilisé des approches de modélisation pour répondre à des questions géométriques, et nous les avons appliqué à des problématiques forestières. Il en résulte un pipeline de traitements cohérent qui, bien qu'illustré sur des données forestières, est applicable dans des contextes variés.

Mots clés : Modélisation géométrique, Formes tubulaires, Normales, Transformée de Hough, Contours actifs, Nuage de points, Bruit, Occlusion

Abstract

The potential of remote sensing technologies has recently increased exponentially: new sensors now provide a geometric representation of their environment in the form of point clouds with unrivalled accuracy. Point cloud processing hence became a full discipline, including specific problems and many challenges to face. The core of this thesis concerns geometric modelling and introduces a fast and robust method for the extraction of tubular shapes from point clouds. We hence chose to test our method in the difficult applicative context of forestry in order to highlight the robustness of our algorithms and their application to large data sets.

Our methods integrate normal vectors as a supplementary geometric information in order to achieve the performance goal necessary for large point cloud processing. However, remote sensing techniques do not commonly provide normal vectors, thus they have to be computed. Our first development hence consisted in the development of a fast normal estimation method on point cloud in order to reduce the computing time on large point clouds. To do so, we locally approximated the point cloud geometry using smooth ”patches“ of points which size adapts to the local complexity of the point cloud geometry.

We then focused our work on the robust extraction of tubular shapes from dense, occluded, noisy point clouds suffering from non-homogeneous sampling density. For this objective, we developed a variant of the Hough transform which complexity is reduced thanks to the computed normal vectors. We then combined this research with a new definition of parametrisation-invariant active contours. This combination ensures the internal coherence of the reconstructed shapes and alleviates issues related to occlusion, noise and variation of sampling density. We validated our method in complex forest environments with the reconstruction of tree stems to emphasize its advantages and compare it to existing methods.

Tree stem reconstruction also opens new perspectives halfway in between forestry and geometry. One of them is the segmentation of trees from a forest plot. Therefore we also propose a segmentation approach designed to overcome the defects of forest point clouds and capable of isolating objects inside a point cloud.

During our work we used modelling approaches to answer geometric questions and we applied our methods to forestry problems. Therefore, our studies result in a processing pipeline adapted to forest point cloud analyses, but the general geometric algorithms we propose can also be applied in various contexts.

Keywords : Geometric modelling, Tubular shapes, Normals, Hough transform, Active contours, Point cloud, Noise, Occlusion

Remerciements

Cette thèse s'est effectuée entre le Centre d'Applications de de Recherches en Télédétection (CARTEL) de l'Université de Sherbrooke au Canada (Québec) d'une part, et l'équipe G-Mod du Laboratoire des Sciences de l'Information et des Systèmes (LSIS), UMR 7296, d'Aix-Marseille Université d'autre part. J'aimerais remercier ces deux universités qui ont permis de financer cette thèse, ainsi que ces deux équipes d'accueil pour leurs contributions et l'accès aux établissements de recherche.

J'exprime mes remerciements les plus sincères à Alexandra Bac et Richard Fournier. En premier lieu pour m'avoir guidé tout au long de mes travaux grâce à leurs commentaires et encouragements, mais aussi pour le bon temps et les discussions intéressantes partagés. Ce travail n'aurait pas été possible sans leurs conseils et leur support.

J'aimerai remercier Marc Daniel, chef de l'équipe G-Mod (LSIS) qui a bien voulu prendre en charge cette thèse et m'a donné l'opportunité d'étudier dans son laboratoire.

Je tiens aussi à remercier Cédric Véga, Alexandre Piboule et Jean-François Côté qui ont suivi mon travail avec attention et m'ont été de précieux conseils.

Je remercie profondément les membres du CARTEL pour avoir fait partie de mes discussions et de mes avancements, ainsi que pour leur gentillesse et leur convivialité. J'adresse également mes remerciements aux membres de l'équipe G-Mod pour avoir créé un environnement plaisant et stimulant.

Je suis également heureux d'exprimer mes remerciements à mes amis pour avoir été une source de discussions scientifiques et d'inspiration, mais aussi pour leur support inconditionnel à chaque moment de ce doctorat. Merci à Caroline et Brice pour les moments partagés à Sherbrooke, merci à Corentin, Célia, Arnaud et Pierre pour la qualité du temps passé à Marseille.

Finalement, je remercie particulièrement ma sœur Églantine, et mes parents, Denise et Jean-Louis, pour m'avoir toujours soutenu dans mes choix, et m'avoir encouragé tout le long de mes travaux. Sans vous cette thèse n'aurait pas été possible.

Table des matières

Liste des figures

Liste des tableaux

Introduction

Ubi materia, ibi geometria.
Where there is matter, there is geometry.

Johannes Kepler

Matter forms objects, therefore where there are objects, there is geometry. Any object can be described by its usage, its physical and chemical properties or its mechanics. However, it is primarily the *shape* that defines an object. The *shape* of an object is the human representation of its geometry, which includes, the object position, size, volume, surfaces and features. When facing several objects, it is indeed their geometry that first differentiates them. Reciprocally, given a geometry, context and human knowledge are introduced to identify the object.

Besides their practical usage, human beings interact with objects in two ways. The first type of interactions is practical, we produce objects to help us and entertain ourselves. The first step for designing an object is to define its geometry: the designer integrates geometric constraints to generate the *shape* of the future object and models surfaces in accordance. The second type of human-objects interactions aims at increasing our knowledge about our environment and history by studying existing objects (e.g. astronomy, archaeology, medicine, engineering, etc.). The automatic study of an existing object requires first to acquire it via remote sensing techniques, which nowadays mostly generate point clouds. Thus, the studies of real objects or environments require adapted methods to process these data and to generate accurate models that approximate closely the reality for precise measurements. If no *a priori* about the acquired object is available, unsupervised reconstruction can be applied to obtain a geometric model and retrieve its geometry. Further analyses of the model may then lead to the recognition of the object, in which case the operation is defined as classification. However, when prior information is accessible, it is possible to recognize an object from a data set and reconstruct it in a supervised manner. This pattern recognition procedure conducts to the obtention of precise measurements.

We illustrated here four different fields of geometric modelling: *designing* an object, *analysing* a model, *recognizing* a shape and *reconstructing* surfaces. In this thesis, we will address the problem of the recognition and the accurate reconstruction of a generic shape from point clouds: in our study, we focus on tubular

shapes, which are an abstraction of a large set of real objects. Depending on the environment, they can represent vessels, poles, pipes, and any surface of revolution around a central curve. In our study, we privileged the application of our methods in the context of forestry where the geometry of trees is predominant in numerous applications.

1. Object recognition and reconstruction from point clouds

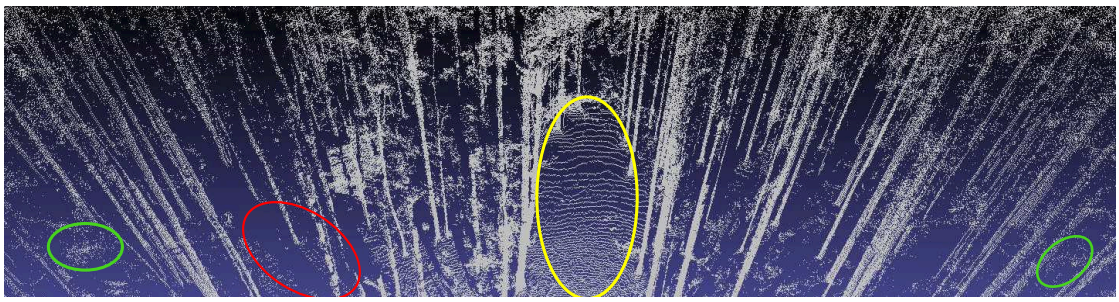


Figure .1. – Illustration of the major challenges encountered in point cloud processing: a terrestrial LiDAR forest plot acquisition containing several tens of billions of points suffers from non-homogeneous sampling density (yellow), noise (green) and occlusion (red).

Along with the development of modern remote sensing, point cloud processing has been studied for the last decades and has become an entire field of study. However it still encounters some major challenges when facing large data sets, non-homogeneously sampled point clouds, occlusions and noise (Figure .1). The size of the data sets constitutes an important limitation for efficient processing and often results in time consuming algorithms. Also the non-homogeneous sampling density increases the difficulty of choosing an appropriate scale for geometrical analyses. Finally, occlusion and noise make the recognition and reconstruction problem difficult: noise scrambles the shape to be reconstructed and generates unwanted artefacts, whereas occlusions produce incomplete and discontinuous shapes.

In this thesis, we aim at extracting a class of objects from point clouds while alleviating the impact of data imperfections. One of our main objectives was to develop robust methods for extracting and accurately reconstructing tubular-shape-based objects in highly occluded and noisy data. Our proposal contains three major points. Our first focus was to propose a fast algorithm to enrich the point cloud with geometric information in the form of normal vectors and

curvature indicators. We achieved this goal by dividing the point cloud into large smooth patches that can be handled efficiently. By doing so, we reduced the impact of data size on the computational efficiency of normal estimation. As a second contribution, we exploited these additional information to design a new algorithm for the reconstruction of tubular shapes which proved its robustness in spite of the data complexity. Indeed, we validated the proposed reconstruction with a set of more than eighty tubular objects with associated measurements. As a third contribution, we use the reconstructed shapes as a starting point to assign each non tubular part of a point cloud to a detected shape and provide a full segmentation of a point cloud into objects.

2. Solutions and motivations

The identification of the point clouds defects led us to determine appropriate methodological choices and select the relevant geometrical tools. During the estimation of normal vectors, we reduced the complexity of surface approximation by using simple explicit surface models and space partitioning structures. Likewise, we decided to model the tubular shapes as parametric surfaces since they are more adapted to pattern recognition than implicit models. By doing so, we also avoided the use of mesh models that are costly to generate on large data sets and for which construction would be more sensitive to noise and occlusions. We rather took advantage of the robustness of parametric pattern recognition approaches for the reconstruction of tubular shapes. Finally, in ongoing works, we mix morphologic and geometric knowledge to separate the objects present in a point cloud and achieve an efficient segmentation of the data.

3. Application to forestry

Nowadays, remote sensing technologies are able to capture the geometry of forest plots, and foresters have great hopes to use them for automatic measurements impossible to obtain otherwise. With point clouds derived from Terrestrial Laser Scanners (TLS), new geometric questions arise. In particular, the extraction and reconstruction of tubular shapes fits perfectly within forestry studies since tree stems and branches are commonly modelled as tubular objects. In addition, point clouds acquired in natural forest environments are among the most affected by the data limitations: TLS generate large point clouds that often need to be registered, producing even larger data sets. Acquisitions in natural forest

environments contain numerous occlusions caused by vegetation, branches and leaves. They are also greatly affected by noise at the branches extremity and in the upper part of trees. As a result, these data sets are particularly well suited to our problem of pushing the robustness of object reconstructions from imperfect point clouds. They provide us a great feedback on the abilities of our algorithms as well as an access to a solid validation with ground truth.

Our solutions for the extraction and reconstruction of tubular shapes geometry and remote sensing of forestry feed one another. The geometric modelling uses natural forest data sets to be confronted to difficulties and prove its robustness. For its part, forestry greatly benefits from the accurate modelling of tree stems thanks to the automatic measurements that solve the data limitations. Moreover, our study opens new questionings that combine remote sensing, geometry and forestry. For example, how can computer graphics and geometric modelling provide new measurements currently inaccessible via remote sensing techniques? How does precise perception of the natural environments geometry will change our understanding of Nature? And a major question is currently asked to the forestry community that is: are TLS measurements more precise than traditional human measurements?

4. Thesis outline

This thesis addresses the recognition and extraction of tubular shapes from unorganized point clouds. Generic solutions are proposed and their results are mainly illustrated on forest point clouds because of the complexity of the data. In addition, we laid the foundations for a novel segmentation of forest point clouds.

First, in chapter I, we briefly review the different needs of forestry that led us to select this field of application for our methods. We also discuss the imperfections of the data sets to show how they match the criteria for the evaluation of our methods before detailing how our specific objectives of geometric reconstruction closely matches forestry. We based our methodological choices on the analysis of the strengths and weaknesses of existing solutions. Therefore chapter II provides the elements for understanding our choices in an overview of the geometric tools and approaches available for point cloud processing. We present the solutions we developed in chapters III to VI. We addressed the issue of fast normal vector field estimation on unorganized point clouds in chapter III before demonstrating in chapter IV how normal vectors can be used to accelerate and enhance the robustness of tubular shapes reconstruction. A complete evaluation of our tubular shape extraction algorithm, including validation against ground

truth and comparison to other approaches, is then provided in chapter V. Chapter VI includes a proposal for object segmentation derived from previously detected shapes. Finally we discuss our methods and our results previous to concluding on the accomplished work and the remaining challenges of point cloud processing in the last chapter of this thesis.

I. Remote sensing of forestry and derived point clouds

Introduction

Today, not only forest resources and forest management constitute important economical and industrial matters, but forestry also plays a major role in the understanding of the future ecological changes. The establishment of sustainable managements of forest environments hence appears to be decisive in many aspects. Consequently, forests are monitored and measured in order to tackle the different issues related to natural resources. However collecting data is a tedious task because of the huge areas covered by various types of forests. Therefore, foresters employ remote sensing techniques to acquire information at large scales. Particularly, Terrestrial Laser Scanners (TLS) (in the form of terrestrial LiDAR, or t-LiDAR) have been introduced to study forests on a plot level. These acquisitions result in dense point clouds that need to be processed to extract meaningful knowledge. Geometry and forestry meet in this context and pursue the same goal: offer alternatives to tedious manual measurements and increase the accuracy of forest measurements.

The main objective of this thesis (robust extraction of tubular shapes from point cloud) is well suited for accurate tree reconstructions from point clouds. Therefore forestry became the main application of our developments: we chose to test our algorithms against t-LiDAR point clouds acquired in natural forest environments. In this chapter we detail the needs of forestry science to better apprehend the pertinence of applying geometric modelling to forest data sets. We also provide the elements of remote sensing techniques necessary to understand the data we will manipulate all along our work (forest point clouds include specific limitations of t-LiDAR acquisitions). Finally, with the fine understanding of this particular applicative context, we present the main objective of our thesis.

1. Forestry science: needs and applications

> *Forestry is the science and craft of creating, managing, using, conserving, and repairing forests and associated resources to meet desired goals, needs, and values for human and environment benefits. Forestry is practiced in plantations and natural stands. The science of forestry has elements that belong to the biological, physical, social, political and managerial sciences.*"
>
> Wikipedia, June 2017

1.1. Definition of forestry science and applications

Forestry is the science of studying forests environments, including soil topology and characteristics, vegetation and wildlife. For example it addresses questions relative to the forest ecosystem like the interactions between soil, plants and animals. As a major forest element, many efforts are put in the understanding of trees: their life, their development and the interaction with their environment. Therefore, forestry is designed to fulfil several purposes related to botanic and ecology.

In the past, forestry led to fundamental discoveries in medicine, agriculture and history. Nowadays this science covers crucial issues since many forests are key elements of local and worldwide developments. Indeed, with specific flora and fauna, regional landscapes and forests constitute an important part of the cultural heritage: the UNESCO World Heritage forest sites now have a total surface area of over 75 million hectares (1.5 times the surface area of France) and represent over 13% of all IUCN (International Union for Conservation of Nature) category I-IV protected forests worldwide.

Forestry knowledge has direct applications in resource management, safety, industry and sciences. Forest inventories are necessary for efficient site planning, successful urban integration and the evaluation of wood production in plantations. Continuous monitoring also reduces the risks of fire hazards and allows to make good usage of ecosystem services. In life sciences, the understanding of plants development is of major importance, and the derived growth models and allometric models find a wide range of applications.

Allometric models focus on finding relationships between the size and the shape of an object. These models are out of our field of study but intervene in chapter V, therefore we introduce here some coarse elements for understanding allometry theory in forestry. Different types of allometric models have been proposed through the years. For example, it has been proved that tree taper can be well approximated from measurements of tree height h and stem diameter at breast height dbh (diameter of the stem 1.30 m above ground, or DBH) via the following equation [183]:

$$d^2(h) = dbh^2 \frac{H-h}{H-h_{dbh}} \left(\frac{h}{h_{dbh}}\right)^{2-\gamma} \tag{I.1}$$

where $d(h)$ is the estimated diameter at height h and dbh is the stem DBH. H and h_{dbh} are respectively the total height of the tree and the height at which DBH was estimated (may differ from 1.30 m depending on the logistic). Finally γ is a specie specific constant. Other models predict a structural or functional variable Y in the form of a power functions of its mass M [86]:

$$Y = Y_0 \cdot M^b \tag{I.2}$$

where Y_0 is an initial measure of the variable Y. Coupled with forest monitoring, these models are used to evaluate and predict forest evolution over time, the impact of climate change or wood production for industrial usages.

1.2. Forestry in practice

As an experimental science, forestry requires to collect data from the field. Hence forest plot inventories have been set up to measure the important characteristics of trees in their natural environment. These measurements are essential for sustainable practices, experimental and operational surveys such as forest management [86], allometry [66] or forest ecology [80]. For a better comprehension of forestry needs, we now define the notion of forest plot. In the applicative context of our thesis we dealt with two key variables measured in forest plots: stem diameter at breast height and stem taper. Therefore we also define these variables and the corresponding measurement tools during field campaigns.

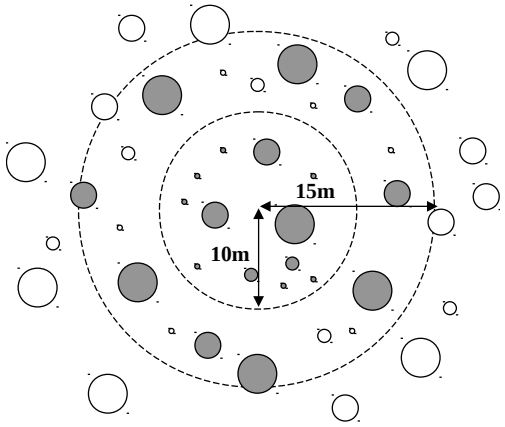


Figure I.1. – All trees are measured in a 10 m circular zone around the plot centre whereas only large trees are measured in the range of 10 m to 15 m.

Definition of forest plot

A forest plot is the unit of forest sampling for measurements. In France forest plots are defined as a circular area with a 15 m radius [88]. In Québec, forest plots are defined as a 400 m^2 circular area, corresponding to a 11.28 m radius [51]. Forest plots include two distinct zones where different types of tree are measured (Figure I.1). In France, only trees with a stem diameter greater than 27.5 cm (31 cm in Québec) are measured in the range of 10 m to 15 m (11.28 m to 14.10 m in Québec) while all trees with a stem diameter above 9 cm are measure in the central part of the plot.

Definition of stem diameter at breast height and measurement

The stem diameter at breast height, also denoted as DBH, is crucial in numerous allometric models. It is the diameter of tree stem, including bark, 1.30 m above ground (note that depending on the field situation, the measurement protocol may differ, see Figure I.2).

During field campaigns, DBH is measured by forest operators using diameter tapes or tree calipers (Figure I.3). Even though these instruments are convenient and commonly employed, manual DBH measurements with such tools generate errors. Firstly, it is important to note that the measurement is objective: the measurement height and the instrument manipulation depend on the operator and his experience. Secondly, these measurements are subjects to errors [10, 100, 103]. Finally, additional errors may appear while reading the measure and storing it into a spreadsheet. Foresters are aware of these weaknesses and they defined an implicit range of acceptable error in DBH measurements: around 2 cm depending on the stem diameter.

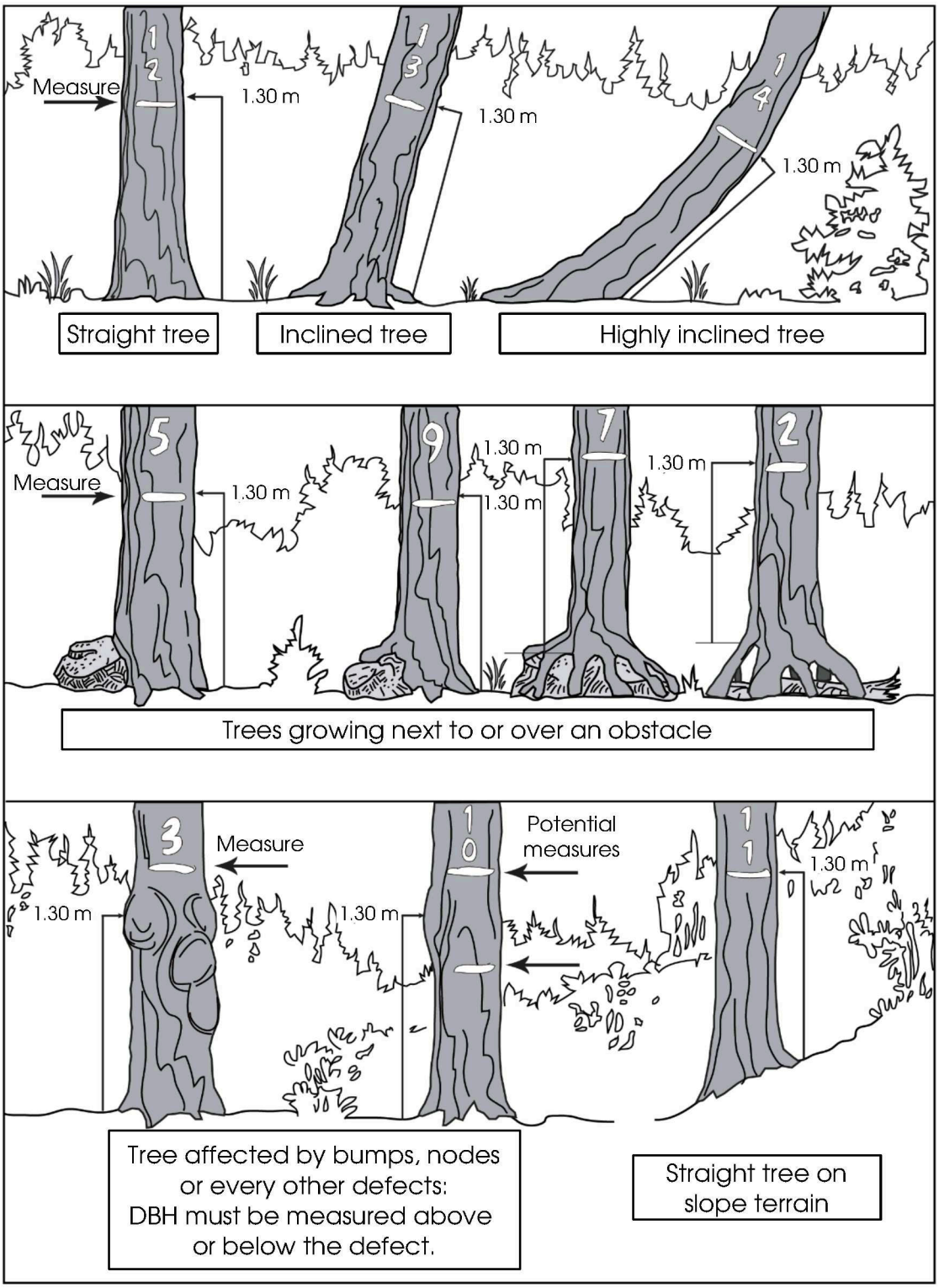


Figure I.2. – Different situations leads to different measurements of DBH. Adapted from [51].

(a) Diameter tape. ©Christina Waitkus. (b) Tree caliper.

Figure I.3. – Measurement of DBH using different tools.

Definition of stem taper and measurement

Stem taper is the decreasing of stem diameter with height. It is an important measure in wood industry and in the development of accurate growth models and allometric equations.

Stem tapper can not be consistently measured during forest inventory campaigns with non-invasive methods because of logistic constraints. Therefore, stem taper is usually measured from destructive protocols: measured trees are cut and pruned before being cut into logs. Then the logs volume or diameter are easily measured by operators. The destructive protocols are costly, and require a lot of time and human resources. Therefore, despite the lack of data, stem tapper is usually not measured during field campaigns.

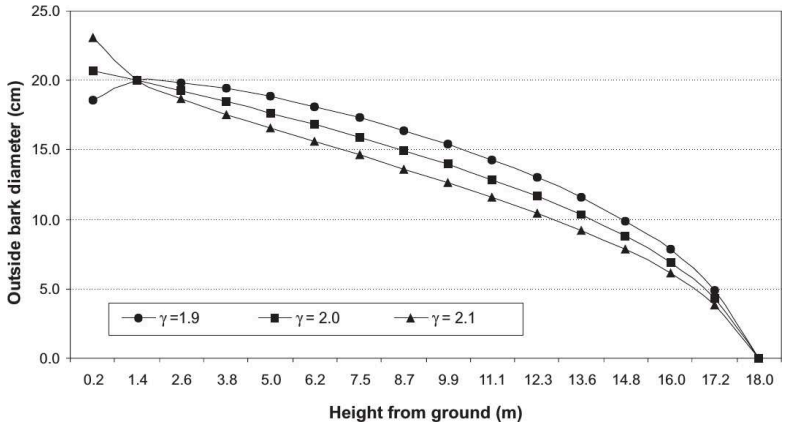


Figure I.4. – Example of stem taper derived from Equation I.1 considering a 18 m tall tree with a 20 cm DBH and different values of γ. Adapted from [183].

1.3. Needs and evolution of forest science

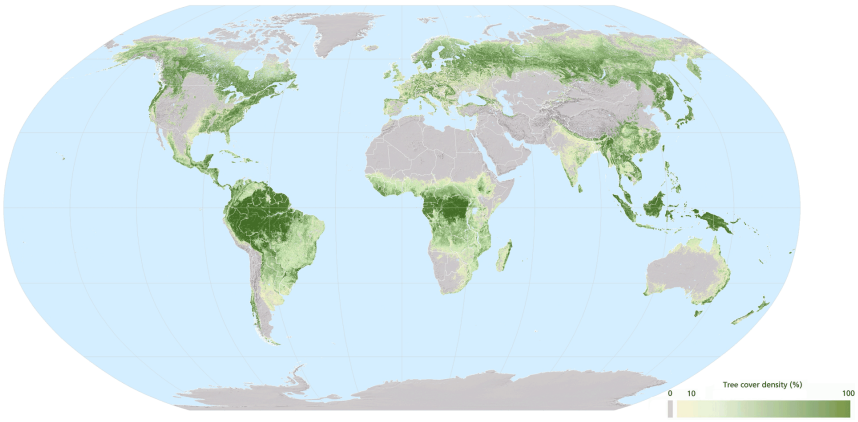


Figure I.5. – World map of tree cover density. Map derived from [76].

With specific needs, forestry has naturally evolved to integrate additional techniques. We now identify the needs of forestry that led forestry and geometric modelling to meet.

Firstly, forests cover a large part of the lands (Figure I.5), hence forestry aims at collecting more data around the globe to provide a global understanding of the ecological mechanisms. Indeed, with large scale studies comes a better perception of the geolocalisation and distribution of forest types and their interactions. Secondly, traditional forest measurements are expensive in time and human resources. Therefore, their automation would increase the number of field campaign and provide supplementary data sets to support forest modelling. Also, as mentioned previously, on-field measurements are subjective, limited by the logistic and include errors. Hence forestry needs to increase the data accuracy and aims at finding objective measurement protocols. Finally, several forestry questions include variable measurements inaccessible to this day and the integration of new variables in forest models will enhance existing results. These needs are the reason why forestry tends to integrate remote sensing techniques in practical applications.

2. Remote sensing of forests

> *Remote sensing is the acquisition of information about an object or phenomenon without making physical contact with the object and thus in contrast to on-site observation. Remote sensing is used in numerous fields, including geography, land surveying and most Earth Science disciplines (for example, hydrology, ecology, oceanography, glaciology, geology) ; it also has military, intelligence, commercial, economic, planning, and humanitarian applications.*
>
> Wikipedia, June 2017

Remote sensing of forestry is a very convenient way to meet the needs of forestry domain. It is efficient, provides large data sets with numerous acquisitions (different techniques, different resolutions, different content) and gives access to additional objective and reproducible measurements.

2.1. Remote sensing

Remote sensing incorporate numerous solutions for the acquisition and measurement of objects at various scales. It was first introduced in forestry to allow studies at scales too large to be observed on the field by humans with non invasive protocols: as early as the 1930's, H. E. Seely used aerial photographies to estimate timber volume and identify tree species [147].

The sensors used in remote sensing can be classified in different groups depending on their characteristics and the type of data collected. For example passive sensors only collect information emitted from their environment (e.g. traditional cameras) whereas active sensors emit a signal pulse and analyse its reflection (e.g. sonars). Note that sensors can be designed to be monochromatic (sensitive to a single wavelength), multispectral (sensitive to a few wavelengths), or hyperspectral (sensitive to hundreds of contiguous wavelengths). Sensors are also characterized by their position. For example, a camera can be mounted on a tripod on the ground, on a satellite or on an underwater support. Remote sensing instruments generate different output. Sensors commonly acquire one of the following data types: 1D signal intensity over time, image and multispectral image, and point clouds. Finally, sensors can be characterized by their resolutions and the different scales of data acquisition.

The sensors characteristics guide the choice of the appropriate instrument for each applicative field. In forestry, aerial photos and radar acquisitions were predominant. But in the last decades, laser scanners have been broadly used with success. During our work, we mainly processed data derived from laser scanners and we showed the relevance of our developments in the applicative context of forest measurements. Therefore, let us now present the laser scanning technology in order to emphasize the pertinence of this solution for forestry purposes, and to illustrate the challenges associated to the output data.

2.2. Laser scanning technology

The recent LiDAR technology (Light Detection And Ranging) is based on a simple principle: an active sensor (laser scanner) emits a light pulse in the form of a laser beam that is backscattered to the instrument by its environment. Although this technology finds atmospheric usages, they are out of the scope of our work and. Therefore in the following we only consider the instruments dedicated to geometric remote sensing and detail the challenges associated to LiDAR data processing.

Geometric laser scanners

The laser scanner manipulated during this thesis generate a point cloud representing their environments. Considering the sensor as the origin of an Euclidean 3D space, a point is recorded in the direction of the laser beam at distance where the signal is reflected by an object. Note that depending on the instrument, the analysis of the reflected signal can generate several points in the direction of the pulse. These instruments are called multiple return laser scanners and are derived from full waveform analyses. Two types of laser scanners are dedicated to geometric acquisitions: *phase-shift* and *time-of-flight* instruments. *Phase-shift* sensors analyse the phase of the backscattered signal to deduce the distance from the sensor to the object reflecting the laser beam. On the contrary, *time-of-flight* deduce this information from the time delay separating the laser pulse and its backscattering from an object.

Supports of geometric laser scanners

Geometric laser scanners can be mounted on various supports (Figure I.6). The applicative context takes advantage of the strengths of the selected support. Airborne scanners (ALS) are mounted on planes and are able to cover large areas but provide a low resolution point cloud. Laser scanners can be fixed on drones for higher resolution aerial acquisitions but the study area is then limited. Mobile

(a) (b)

Figure I.6. – Principle of LiDAR scanners. Airborne sensors scan their environment from above by sweeping a laser beam while moving (a). The terrestrial instrument rotates vertically and horizontally while emitting laser beams in order to generate a point cloud of its surrounding environment (b).

laser scanners (MLS) are mounted on vehicles to scan large areas from ground locations, however the scanning environment is limited by logistic. Terrestrial laser scanners (TLS, or t-LiDAR) are also located on the ground and provide high resolution point clouds of their surrounding in a limited range. Finally hand-handled sensors are manipulated by walking operators and hence consume a large amount of time and human resources.

From the listed supports, it appears that mobile and hand-handled are not adapted to fast forest monitoring. On the contrary, airborne, drone and terrestrial instruments are well suited for forest studying. However, they provide different informations about forests:

ALS are mainly used to estimate forest attributes related to tree heights, tree density estimation and terrain topography. Indeed, the aerial point of view is ideal for scanning the upper part of trees and the large laser beam employed by ALS allows to estimate first and last signal return corresponding to tree canopy and soil. However, the angle of acquisition, the low resolution of these instruments and the signal attenuation in foliage limits the information about trees structure, including tree stem and branches where only a few points are detected.

Drones are a new support for LiDAR instruments. These instruments are well suited for coarse global analyses of forest vegetation. They are able to produce point clouds describing tree foliage as well as tree stems and branches. However, the resolution of drone-mounted sensors can not describe accurately the geometry of woody parts and hides major details.

Instrument specification	Example value	Quality impacted
Wavelength	905 nm (near infrared)	Signal reflectance and noise due to solar radiation
Maximum range	120 m	Farthest object scanned
Measurement speed	up to 976,000 pts/sec	Points regularity and accuracy
Range error	± 2 mm	Points accuracy
Ranging noise	0.6 mm at 10 m	Points accuracy
Angular resolution	up to 0.009°	Point density
Field of view (vertical/horizontal)	$305^\circ / 360^\circ$	Missing data
Beam divergence	0.011°	Noise and outliers

Table I.1. – Instruments specifications and impact on data quality. Example values correspond to the Faro Focus 3D© 120 scanner.

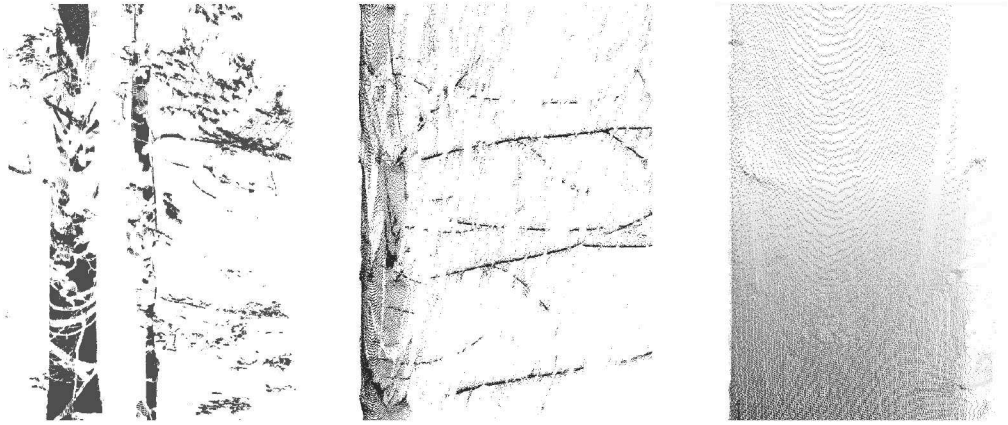
TLS are the pendants of ALS since they are well adapted to studies on trees structure. The high resolution acquisitions combined with ground location provides an accurate representation of trunks and branches in the lower part of the trees. On the contrary, the laser signal can not penetrate the upper part of the tree because of the presence of foliage, and hence TLS poorly describe the canopy elements.

TLS are therefore by far the preferred option for forest studies focusing on tree structure estimation including stems DBH and taper measurements. Moreover, TLS instruments have evolved and can now be easily transported and offer fast acquisitions.

2.3. Terrestrial LiDAR data

Point clouds acquired with TLS instruments suffer from several limitations. These limitations are amplified in the context of forest acquisitions. We now give insights about TLS acquired point clouds to better apprehend the challenges of developing robust point cloud processing algorithms for such data sets.

The quality of TLS point clouds is affected by two major elements. They are closely related to the instruments characteristics detailed in Table I.1. The environment also greatly impacts the resulting point clouds. Indeed, point clouds acquired in forest environments highly suffer from three limitations (Figure I.7).



(a) (b) (c)

Figure I.7. – Illustration of TLS data limitations in forest environments. Forest point clouds suffer from occlusion (a), noise (b) and non-homogeneous sampling density (c).

Occlusions are due to understorey vegetation, branches and foliage. Forest point clouds also contain a high amount of noise due to the laser beam hitting multiple surfaces (e.g. two branches). Finally, TLS acquisitions generate non homogeneous sampling density because of the spherical geometry of the sensor.

3. Thesis objectives

The objective of our thesis is the development of a robust tubular shape reconstruction from occluded noisy point clouds and its application to tree stem reconstruction. This general goal has been split into 3 specific objectives.

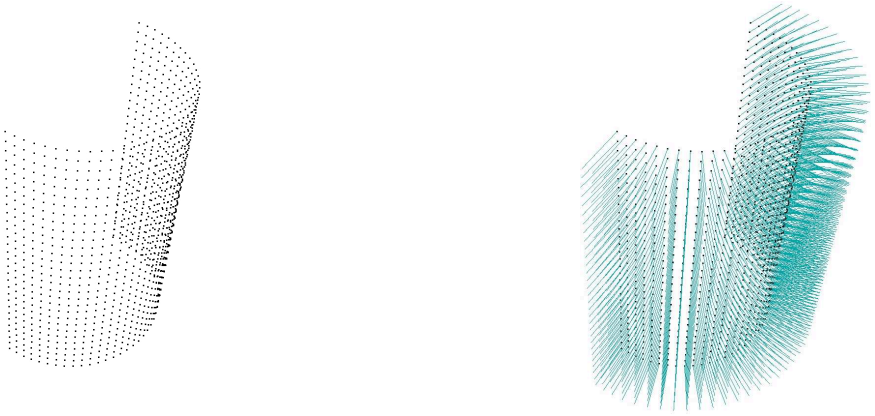
As a prerequisite for further developments, normal vectors are crucial in our pipeline processing. However, the data manipulated during our works contain several tens of millions of points. The estimation of normal vectors on these data sets was time consuming and represented a limitation for the efficient application of our tubular shape extraction algorithm. Our first specific objective therefore consisted in providing a fast normal estimation algorithm on point clouds (Figure I.8).

Our second objective is the core of our work. It was defined to be the development of a tubular shape extraction able to handle the major challenges of forest TLS point clouds (Figure I.9). In addition to the development of the robust methodology, we extensively validated our algorithm on a wide range of data sets

to provide a quantitative evaluation of its robustness.

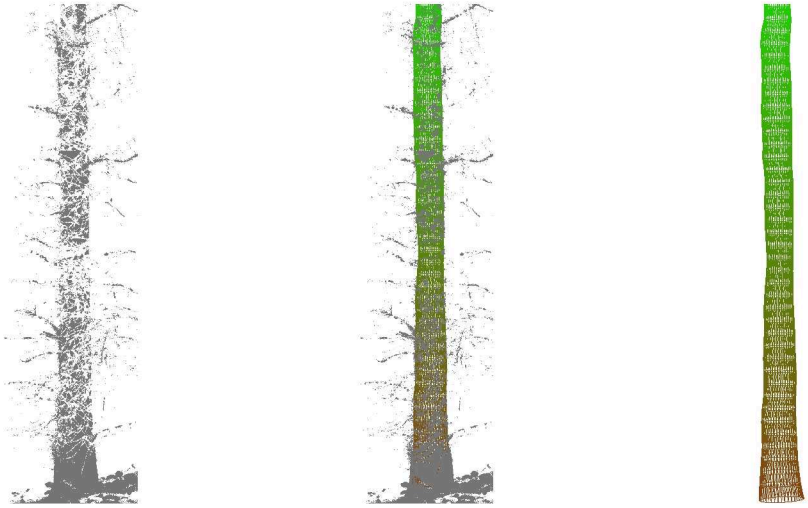
Finally, our third specific objective aimed at using the previously extracted tubular shapes to isolate the trees of a forest plot into separated segments (Figure I.10). The proposed solution is intended to give access to additional forest variables inaccessible to this day and to improve the estimation of tree crown attributes.

We intensively used geometric modelling techniques to fulfil our specific objectives. Each of them feed the other one, forming a consistent modelling pipeline which can be used in a wide range of applications, even though we illustrate the results in a forestry context.



(a) (b)

Figure I.8. – Out first objective is to provide a fast normal estimation algorithm on point cloud.



(a) (b) (c)

Figure I.9. – The second specific objective of this thesis addresses the robust reconstruction of tubular shapes from point clouds and its application to TLS-scanned tree stem reconstruction.

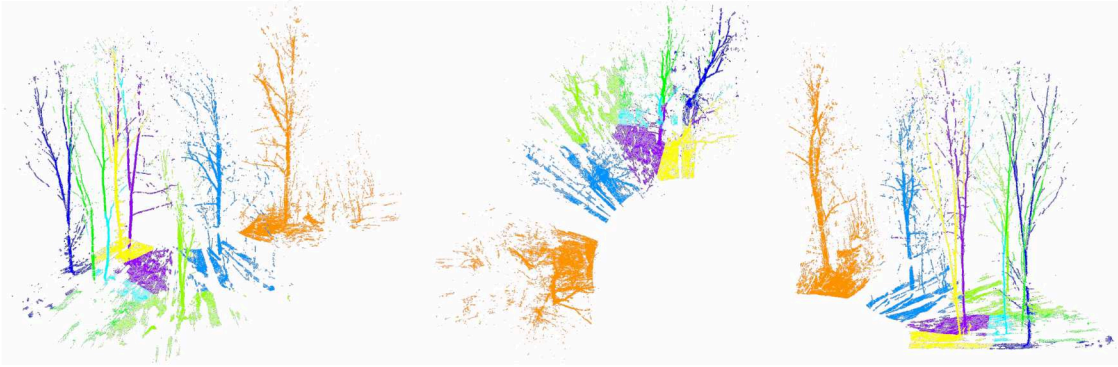


Figure I.10. – Our third specific objective is to segment each tree of a TLS-scanned forest plot.

II. Point cloud and geometric modelling

> Geometric modeling is a branch of applied mathematics and computational geometry that studies methods and algorithms for the mathematical description of shapes.
>
> Wikipedia

Introduction

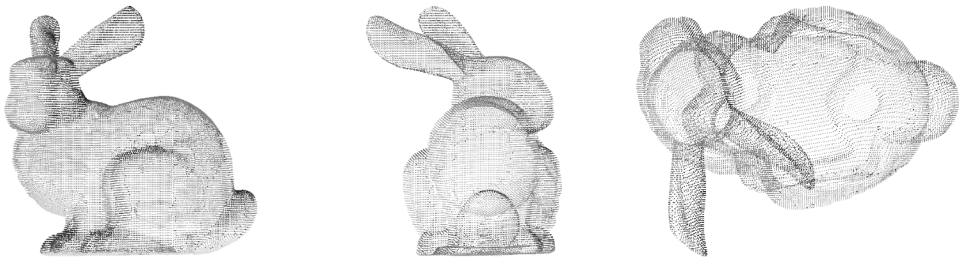
With the recent development of remote sensing technologies (e.g. photogrammetry, laser scanning, depth sensors), point cloud processing has become an integral and independent discipline. This field of study includes various elements: among others, numerous studies have focused on visualisation, segmentation, modelling, pattern recognition, surface reconstruction and data compression. Broadly speaking, point cloud processing consists in extracting information on the content of a point cloud at a medium to high level. For example, surface reconstruction provides a medium information level whereas the identification of an object's components is a high information level that increases human knowledge. Today, point cloud processing is the key of many applications including automatic measurements, reverse engineering and shape analyses. Several approaches inspired from different fields of study can be adopted to perform these tasks. Among others, it is possible to adopt statistical or morphological approaches to apprehend point clouds. For our study, we have chosen to focus on the geometry of point clouds to analyse data and extract the properties of objects scanned by t-LiDARs.

The fundamental goal of geometric modelling for point cloud processing is to recover the geometry concealed by the data points. To do so, it is implicitly assumed that points of the clouds are sampled on surfaces. One of the main approaches in point cloud processing is hence to obtain an accurate representation of the data as a set of surfaces. Indeed, such surface-based representation of point clouds allows the estimation of essential geometric features such as volumes, salient features or normal vectors. Such operations are often non trivial

on point clouds since raw data only provide local and fragmented information on objects: among others, the lack of connection between points requires additional notions for local point cloud analyses. For this reason, organizing the point clouds into adapted data structures is a central element. The different available structures not only give rise to computationally efficient procedures, but they also pave the way for additional families of algorithms.

In order to better apprehend the methods presented later in this thesis, we now introduce the main concepts employed throughout our project, as well as the general notions of geometric modelling related to our object of study. At first, we will present an overview of point clouds and we will illustrate some of the major challenges associated to this type of data. Then we will focus on the geometric models available for expressing the underlying surface of a point cloud. We will also introduce the geometric elements that can be derived from a point cloud for supporting the construction of appropriate geometric models along with the data structures used to achieve efficient computations. Last, we will present some major pre-processing steps that prepares the data for further analyses before we illustrate the existing approaches for modelling a point-sampled object.

1. Point cloud and associated problems



(a) Side view (b) Front view (c) Top view

Figure II.1. – Point cloud representation of the Stanford bunny.

Point cloud processing constitutes the core of our work and we consider t-LiDAR acquisitions as the elementary data handled throughout the different solutions presented in our studies. Hence it is crucial to provide a clear definition of a point cloud and to highlight their characteristics before detailing our developments

Point clouds acquired with T-LiDAR can be very dense, and thus offer a rich and precise representation of a scanned scene. However, when it comes to processing these data sets, numerous challenges arise. Let us discuss these challenges from the definition of a point cloud and the differentiation between *unorganized* and *organized* point clouds.

Unorganized point cloud

Since we only manipulate *unorganized* point clouds in this thesis, in the latter, when no precision is given, the term point cloud will refer to an *unorganized* point cloud. The rigorous definition of a point cloud is straightforward. A point cloud P of n dimensions and size k is a set of k n-dimensional points:

$$P = \{p_1, p_2, \cdots, p_k\},\ p_i = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n \tag{II.1}$$

This definition embeds a large variety of point clouds including, in a large sense, statistical data sets. In our study, we only focus on 3D point clouds acquired from remote sensing techniques, and particularly data derived from t-LiDAR scans. These point clouds have strong geometric semantic as the points are ac-

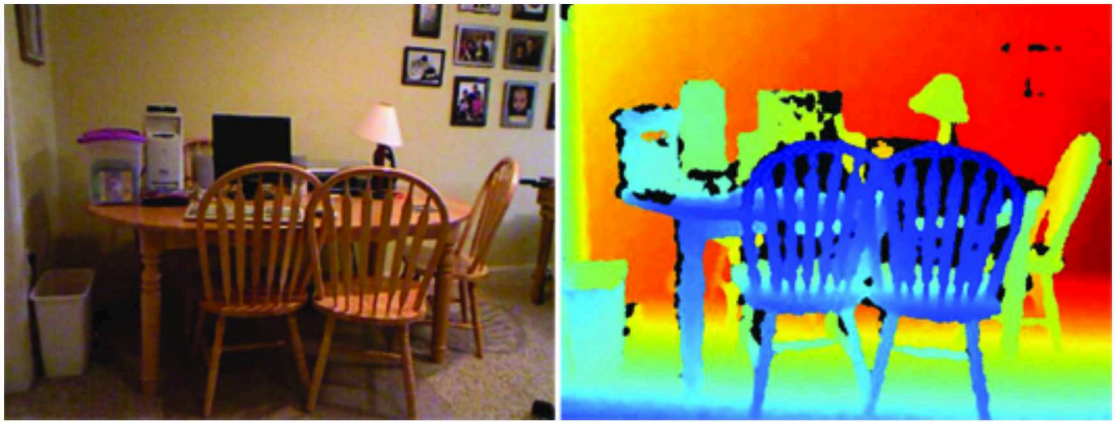


Figure II.2. – Example of *organized* point cloud: the data points are stored in a matrix and are coloured according to their distance from the sensor. Adapted from [146].

tually sampled on surfaces embedded in the classical Cartesian 3D space (Figure II.1). Under certain conditions a 3D point cloud can be regarded as a depth image of a scene, or equivalently, a 2.5D point cloud. This leads to the distinction between organized and unorganized point clouds.

Organized point clouds

In *organized* point clouds, data points can be split into a matrix (i.e. arranged in rows and columns) and visualised as a depth image (Fig. II.2). *Organized* point cloud mostly result from time-of-flight cameras acquisitions where the sensor directly provides matrices of points, or from infrared depth sensors where a regular grid pattern provides the arrangement of points. The main advantage of such point clouds is that the data are structured as soon as they are acquired. Moreover, *organized* data sets benefit from a wide range of image processing algorithms. The matrix structure also determines an intuitive notion of points adjacency derived from image processing and can be used to develop efficient neighbour searches operators [159]. These operators in turn are used to reduce the computational cost of several algorithms and decrease the computing time of point cloud processing.

Whereas *organized* point clouds offer advantages, the solution of regarding a point cloud as a depth image is not often considered in practice since it generates additional issues. As a first issue, obviously not all remote sensing techniques provide *organized* data sets. Even though t-LiDAR acquisitions can be stored according to the spherical coordinate system of the sensor, the transformation from the sensor's spherical coordinate system to the Cartesian plane (the domain of a depth image) is a projection problem widely studied in geography (e.g. Mer-

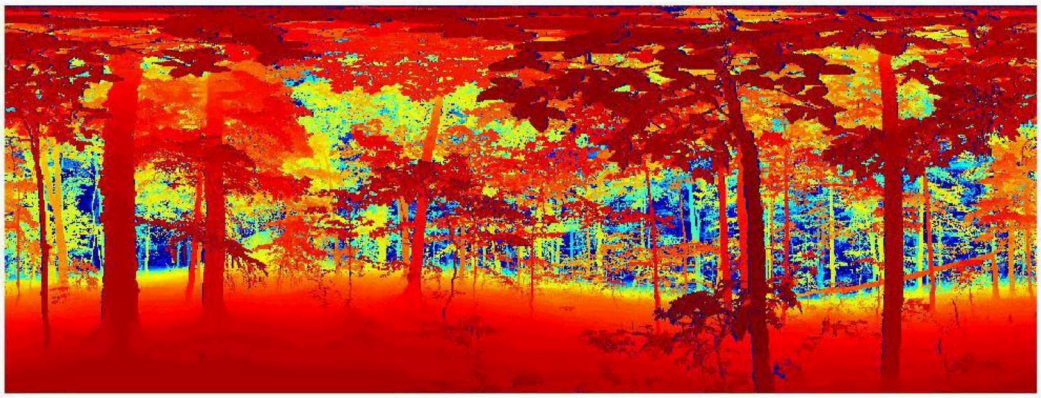


Figure II.3. – t-LiDAR acquisition transformed into a range image (colours correspond to the range). Large distortions appear at the top due to the spherical geometry of the sensor. Adapted from [148]

cator projection). The solution is not unique and solutions generate distortions on the final image (Fig. II.3). Moreover, there is no existing solution for creating an *organized* data set from mobile scanning or photogrammetry. As a second issue, it is not possible to merge two *organized* point clouds into a single unified *organized* one. Hence, the advantages of the data organisation are lost when processing multi-scan acquisitions. Therefore, in the following, we will only consider t-LiDAR data sets as *unorganized* point clouds.

Challenges of point cloud processing

Although the data points are sampled on surfaces, they do not provide easy access to these surfaces. Indeed, raw point clouds are a collection of 3D coordinates and as such, they only provide minimal geometrical information. The absence of relationship between data points disallows the direct estimation of the local geometry which is needed by most processing algorithms. In addition, scanning techniques often generate very large point clouds: t-LiDAR acquisitions usually contain up to several tens of millions of points. The amount of data constitute a challenge for visualisation, storage, and processing computational time. Finally, algorithms have to deal with four major limitations of point cloud data (Fig. II.4): (1) occluded and incomplete data, (2) outliers, (3) noise, and (4) non homogeneous sampling density. Occlusions generate missing data ("holes") of various sizes that disrupt the retrieval of the scanned surfaces geometry. Outliers and noise greatly affects the accuracy of the surface estimation whereas algorithms have to take into account the local sampling density to achieve robust results.

As a raw object of study, point clouds are hence difficult to apprehend and raise a major question: how can we retrieve scanned surfaces from sample points

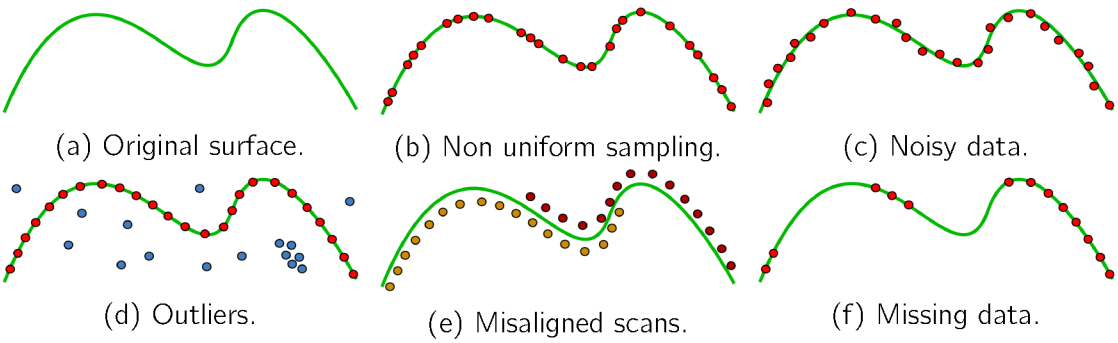


Figure II.4. – Point cloud limitations. Adapted from [17].

and extract pertinent geometrical information that will further let us solve more complex problems? In our work, we particularly addressed the robustness of the tubular shapes reconstruction to provide higher level knowledge about the data content to the user.

2. Geometric models of surfaces

A natural solution to retrieve the underlying surfaces of data sets is to approximate the sampled points with an appropriate surface model. Surfaces representation have been widely studied and there exists more than one model to describe a unique surface. The choice of model greatly influence the point cloud approximation methods and its geometrical analysis. Hence, it is important to be aware of the strengths and weaknesses of each model when facing the problem of point cloud approximation with a surface. In our work we actually different models according to our objectives (fast normal estimation and tubular shapes reconstruction). Hence, before introducing our developments, let us motivate our choices by giving a short presentation of surface models. These models can be organised into two major groups, namely *explicit* and *implicit* models.

2.1. Explicit models

A convenient way to handle a surface is to have a representation in which the exact location of each point of the surface is immediately computable. In other words, these models of surfaces explicitly define all the points on the surface. Actually, we distinguish two types of explicit models, namely parametric models and mesh models. On parametric surface models the points of the surface are

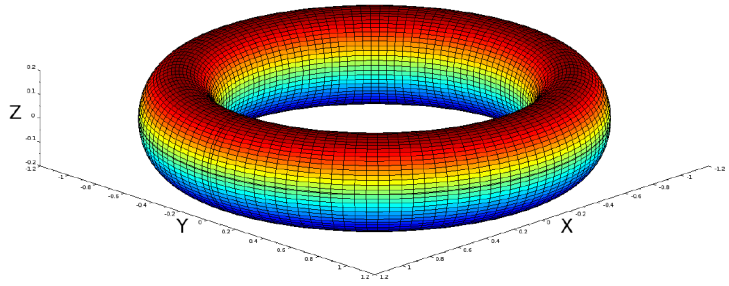


Figure II.5. – A parametric torus of radii R and r is defined by:
$f(u,v) = (R + r \times \cos(u)) \times \cos(v)$,
$g(u,v) = (R + r \times \cos(u)) \times \sin(v)$,
$h(u,v) = r \times \sin(u)$

derived from parametric functions. On meshes, the points of the surface are located on the union of a set of polyhedral faces. Let us now introduce both of these models and underline their characteristics.

2.1.1. Parametric surfaces

Parametric surface models express a relation between a pair of parameters and the points on a surface. Parametric surfaces are the image of a function from a parameter domain $D \subset \mathbb{R}^2$ to $\mathbb{R}^3$.

Definition

A parametric surface S is expressed from a set of parametric equations:

$$S(u,v) = \begin{pmatrix} x = f(u,v) \\ y = g(u,v) \\ z = h(u,v) \end{pmatrix} \tag{II.2}$$

where u and v are the parameters varying in the domain D (Fig. II.5). Hence for a given pair of parameter, the corresponding point on the surface can be explicitly computed. Parametric models embrace a large set of surfaces including folded surfaces and surfaces of revolution.

Point cloud approximation with parametric models via least square optimisation offers an elegant, and often efficient, solution to the problem of surface estimation. Note that algorithms such as the RAndom Sampling Consensus (RANSAC, [63]) can be easily employed to improve the robustness of this procedure. As explicit models, parametric surface models provide a direct access to the points of the surface, which is a strong advantage in many applications (e.g. efficient points projection operators [4]). In addition, parametric surface models are simple to handle and easy to store. In the literature two major types of parametric surface prevail: elevation surfaces and B-Splines patches along with their generalisation as NURBS patches.

2.1.1.1. Elevation surfaces

Elevation surfaces are actually a special case of parametric models where a coordinate (often z) can be solved as a function of the others (x and y). Finding such relation is convenient since it results in a very simple and light solution for studying the geometrical properties of the object.

Definition

Elevation surfaces are simply defined by a Cartesian equation in the form of:

$$z = F(x, y) \tag{II.3}$$

where F is a function from $\mathbb{R}^2$ to $\mathbb{R}$.

Elevation surface models are particularly appropriate when considering elevation surfaces like digital terrain models (DTM) [140]. However, they obviously suffer from their specificity and cannot model folded surfaces. Moreover, elevation surface models are defined globally and do not provide any local control on the surface. Hence, to obtain a slight deformation of a surface the entire definition of the surface model has to be changed. Nevertheless, elevation surface models are well suited to approximate point clouds sampled from simple surfaces using least square optimisation [65], and they provide efficient solutions to several problems including differential geometry computations.

In most applications, the function F is chosen to be a bivariate polynomial or a combination of trigonometric functions since Taylor series and Fourier series have proven to be powerful to approximate a data set. In many works the in-

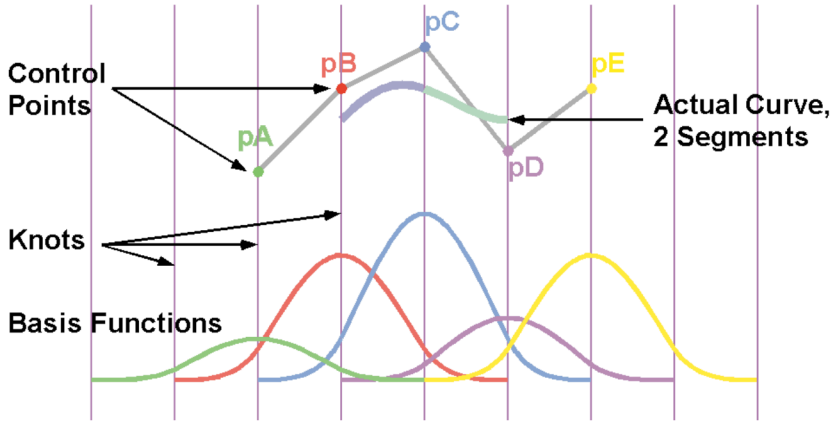


Figure II.6. – B-Splines and NURBS surfaces are defined from a collection of control points and knots based on basis functions.

ability of elevation surface models to represent folded surfaces is overcome by dividing the point cloud into patches patches before finding a local basis (and thus, a natural parametrization of the points) in which points can be approximated with an elevation surface model [145, 149, 171, 198].

2.1.1.2. B-splines and NURBS surfaces

B-splines (Basis splines) surface models are parametric models defined from a set of functions called B-spline functions, and a set of external points called control points (not included in the input point cloud). These models achieve precise surface estimation from point clouds. B-splines models and their generalisation NURBS (Non-Uniform Rational Basis Splines) models are historical and popular set of parametric surface models. They are obtained intuitively as a weighted sum of the set of control points (Fig. II.6).

B-Spline surface definition

The definition of a B-Spline surface from the parameters u and v ($u \in U$ and $v \in V$) involves multiple elements:

— an array of $n_u + 1 \times m_u + 1$ control points $c_{i,j}$, where $0 \leqslant i \leqslant n$, and $0 \leqslant j \leqslant m$
— a vector of $k_u + 1$ knots in the u-direction
— a vector of $k_v + 1$ knots in the v direction
— the degree d_u in the u direction
— the degree d_v in the v direction

with $n_u = k_u - d_u$ and $n_v = k_v - d_v$. With these elements, a B-spline surface S is defined as:

$$S(u,v) = \sum_{i=0}^{n_u} \sum_{j=0}^{n_v} N_{i,p}(u) N_{j,q}(v) c_{i,j} \tag{II.4}$$

where the coefficients $N_{i,p}(u)$ and $N_{j,q}(v)$ of the control points $c_{i,j}$ are computed from B-Spline functions derived from the Cox-de Boor recursion formula [166]:

$$B_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} B_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} B_{i+1,p-1}(u) \tag{II.5}$$

where:

$$N_{i,p}(u) = \begin{cases} 1 & \text{if} u_i \leq u \leq u_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{II.6}$$

NURBS surface definition

NURBS surfaces are a generalization of the B-Spline surfaces where each control point $c_{i,j}$ is associated a weight $w_{i,j}$ [199]. A NURBS surface is defined by:

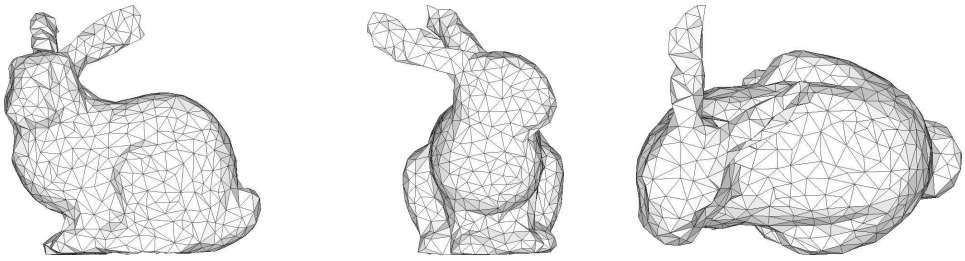
$$S(u,v) = \frac{\sum_{i=0}^{n_u} \sum_{j=0}^{n_v} N_{i,p}(u) N_{j,q}(v) w_{i,j} c_{i,j}}{\sum_{i=0}^{n_u} \sum_{j=0}^{n_v} N_{i,p}(u) N_{j,q}(v) w_{i,j}} \tag{II.7}$$

B-Spline and NURBS surfaces alleviates the limitations of elevation surface models: they can be used to generate folded surfaces, and the introduction of control points gives access to a local management of the surface. Like with elevation models, it is possible to approximate a point cloud by NURBS surfaces using least square optimisation. During this procedure, numerous elements strongly impact the results and have to be taken into account such as the number of control points, their weights and the knots vectors [108]. If the weights and the knots vectors are known *a priori*, the surface approximation is reduce to estimate the control points with linear least squares, otherwise, it requires to solve a non linear least square problem [108]. Similar to elevation surfaces, a point cloud is usually approximated by using multiple local NURBS patches to facilitate computations and further handling [60].

Conclusion

Generalized parametric surfaces and NURBS in particular are broadly used in several domains including Computer Assisted Design (CAD) because they are easy to manipulate and their modification is intuitive. Moreover, it is possible to constraint the geometry of such models (e.g. impose that the surface passes through a specific point, or impose a smooth continuity between patches). They also ensure some smoothness and continuity properties [166]. However, the approximation of point clouds from NURBS appears to be potentially a difficult task: determining knots vectors and weights is a complex issue. Sampled surface are *a priori* not to be modified during point cloud processing, hence we will prefer another surface model.

2.1.1.3. Meshes



(a) Side view (b) Front view (c) Top view

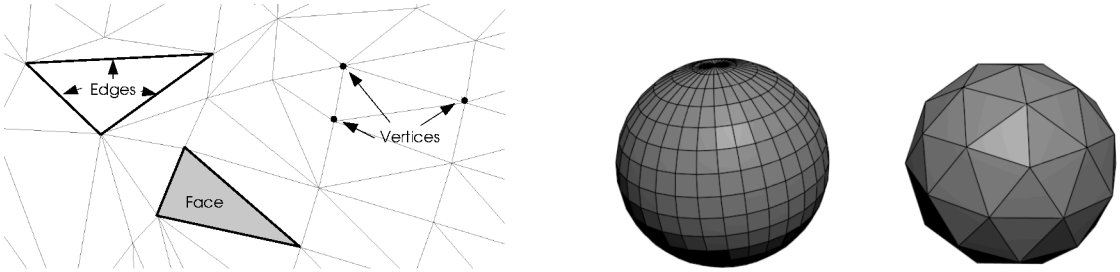
Figure II.7. – Mesh representation of the Stanford bunny.

Modelling sophisticated objects with continuous functions can be a complex task. Therefore, in a large set of applications a discrete representation of a shape as a mesh model is usually preferred (e.g. entertainment, CAD, and 3D printing).

Definition

A mesh model represents a surface as a set of polyhedral faces [28]. More precisely, a mesh M is composed of a set of vertices V connected by a set of edges E that form a set of polygonal faces F:

$$\begin{array}{rcl} M & = & < V, F > \\ V & = & \{v_1, v_2, \cdots v_{n_v}\} \\ E & = & \{e_1, e_2, \cdots e_{n_e}\} \\ F & = & \{f_1, f_2, \cdots f_{n_f}\} \end{array} \tag{II.8}$$



(a) (b)

Figure II.8. – A mesh model includes vertices, edges and faces (a). The most common models are quadrangular meshes (left) and triangular meshes (right) (b).

where each of the n_v vertices $v_i = (x_i, y_i, z_i)$ is a 3D point, each of the n_e edges $e_i = (v_i, v_j)$ is a pair of vertices, and each of the n_f faces $f_i = \{v_{i_1}, v_{i_2}, \cdot, v_{i_m}\}$ is an ordered set of m vertices defining a polygon (Fig. II.8a). Each points on the surface can then be computed using barycentric coordinates. In a general mesh model, there is no restriction as for the type of polygon composing faces. However, we distinguish two special cases commonly used in modelling. Triangular meshes are only composed of triangular faces, providing major geometrical properties. Quadrilateral meshes are only composed of quadrilateral faces (Fig. II.8b). Note that quadrangles may not be planar, and thus many efforts are often done to obtain a triangular mesh model.

Discussion

Meshing point clouds has been widely studied and numerous methods consider the data points as vertices. Meshes are easy to handle, to edit locally with high control of the deformation, and to visualise. They have been broadly used in several fields [28]. Subsequently there are a lot of existing algorithms already available to solve various problems including differential geometry computation. Triangular meshes in particular exhibit specific useful properties when computed from a Delaunay triangulation [41]: uniqueness, dual of the Voronoï diagram, empty circumcircle, efficient computation, etc. However, as a collection of polygons, meshes are only discrete piecewise linear approximation of surfaces. Hence mesh models can not accurately represent continuous smooth surfaces and, unlike other parametric models, the estimation of the differential geometry is not straightforward. Therefore, many works have focused on the computation of normal vectors and curvature of smooth surfaces approximated by discrete meshes [135, 174].

Conclusion

Mesh models offer many conveniences and come with a large set of algorithms that make them appear to be very attractive for surface modelling. However, obtaining meshes from dense point clouds is computationally expensive and non trivial. These models are based on vertices position, therefore they are highly sensitive to noise, and the definition of faces makes them affected by occlusion. Finally the memory footprint of mesh models generated from large point clouds is also an issue.

2.2. Implicit models

(a) Side view (b) Front view (c) Top view

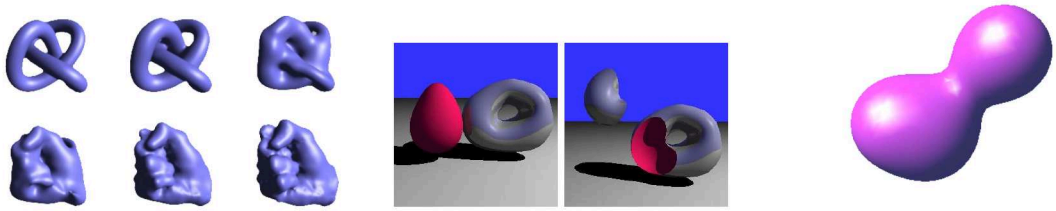
Figure II.9. – Meshed representation of the zero level set of an implicit function defined from the Stanford bunny point cloud.

The family of implicit models is very large. Indeed, any explicit representation of a surface can be expressed as an implicit model while not every implicit surface can be expressed explicitly. This major strength of implicit models is used in many applications including surface reconstruction from point clouds (see Section 5.2) as illustrated on Figure II.9. Implicit models are also used in the entertainment domain since they provide a simple solution to handle interactions between surfaces (e.g. attraction, repulsion), smooth morphing between implicitly defined surfaces, and they facilitate boolean operations such as union and intersection of models (Fig. II.10).

Definition

Implicit surface models are defined as the zero level-set of an implicit function. More precisely, given a scalar field defined by a function $F(x, y, z)$ from $\mathbb{R}^3$ to $\mathbb{R}$, the corresponding implicit surface S is the set of points (Fig. II.11):

$$S = \{p_i = (x_i, y_i, z_i) | F(x_i, y_i, z_i) = 0\} \tag{II.9}$$



(a) (b) (c)

Figure II.10. – Applications of implicit models: (a) smooth morphing (adapted from [196]), (b) efficient boolean operators (adapted from [197]) and (c) smooth objects interactions with metaballs.

Hence, unlike explicit models, there is no direct access to the points lying on the surface model. From the definition, it is obvious that the signed distance function is a perfect example of implicit function modelling a surface [81].

Like NURBS, implicit models are flexible since the implicit function can integrate a set of points in its definition and constraints can be added on the surface model. Our intention in this section is to motivate and justify our methodological choices. Hence, the complete description of implicit models is out of the scope of this document and we only introduce a broadly used class of implicit models: the radial basis functions.

Given a set of n points p_i, an implicit model can be obtained as the following linear combination of (potentially compactly supported) radial function $G(d)$:

$$F(x,y,z) = F(p) = \sum_{i=1}^{n} w_i G(\|p - p_i\|) \tag{II.10}$$

where w_i are weighting factors. Metaballs illustrated in Figure II.10c are an example of radial basis function. They can be achieved from a set of n seed points p_i with associated radii r_i and a basis of function $G(d) = \frac{1}{d^2}$:

$$F(x,y,z) = \sum_{i=1}^{n} G(\|p - p_i\|) = \sum_{i=1}^{n} \frac{1}{(x-x_i)^2 + (z-z_i)^2 + (z-z_i)^2} - r_i) \tag{II.11}$$

Metaballs and their derivates are obviously more suited for CAD and entertainment usages than for surface reconstruction from point clouds. However, when

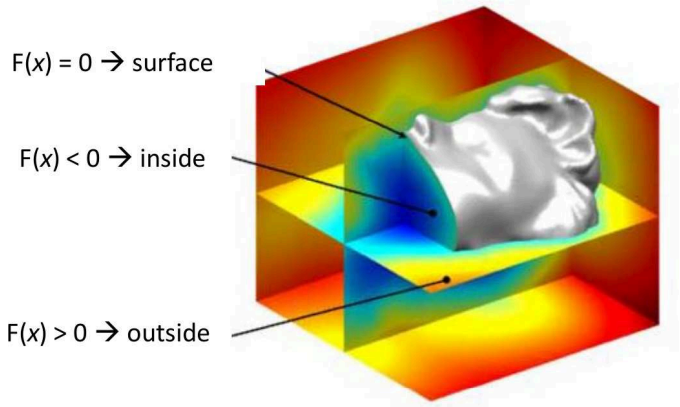


Figure II.11. – An implicit surface if defined as the zero level-set of an implicit equation.

choosing proper radial functions, implicit models exhibit interesting interpolating properties [36] and can be used for surface reconstruction [37].

Discussion

One major strength of implicit models is that the topology of a surface, and potential topology changes during morphing are naturally handled. When it comes to surface approximation, particular attention must be paid to the function F considered. The differential properties of the modelled surfaces are also strongly related to the choice of F. With an appropriate choice we can thus obtain a surface of desired level of smoothness from implicit models. Using these properties, several methods have shown that implicit models are able to approximate a point cloud. In addition, these models do not require a parametrization of data points and differential geometry is well known.

On the side of these advantages, implicit models face limitations. Indeed, they do not provide direct access to points on the surface. Also, the implicit model of a given surface is not unique (e.g. any multiple of a solution is also a solution). Constraints are hence required to disambiguate interpolation and approximation problems. Another drawback of implicit models is their visualization which proves to be costly. Two solutions have been proposed to solve this issue: (1) ray tracing, and (2) the marching cube algorithm and its derived improvement. In the first one, ray tracing algorithms are used to evaluate the intersection of a set of rays and the implicit surface [77]. The intersections can be computed analytically as the solution of an equation involving the ray parametrisation. The marching cube algorithm however is often preferred since it generates an mesh of the surface that can be further analysed or edited for further studies [122]. This algorithm evaluates the values of the implicit function on a 3D lattice in

order to extract an approximation of the iso-surface. Finally, contrarily to parametric representations, it is not possible to implicitly model an open surface (with borders) and it is difficult to restrict the domain of definition of implicit models without prior information.

Conclusion

Many work have shown the potential of point cloud approximation with implicit models and several propositions are found in the literature (see Section 5.2). Whereas implicit models are attractive for smooth surfaces modelling, they are impacted by large occlusion and incomplete data. Also, point cloud approximation with implicit models are affected by outliers and misalignments [17]. Therefore, implicit models are not well adapted to robust shape extraction from complex data acquired in natural forest environments.

2.3. Primitives

Many man-made objects are designed as a combination of common solid shapes such as planes, spheres, cylinders, cones and toruses (Fig. II.12). These shapes are often referred to as primitives and are extensively used in CAD and Constructive Solid Geometry (CSG) applications via boolean operators (union, difference and intersection) [43]. In reverse engineering procedures, the approximation of a data set with primitive surfaces leads to automatic measurements and are used to retrieve the chain of operations to create a shape or to identify construction defects.

Definition

No formal definition of a geometric primitive can be found. However a common notion emerges in the meaning of *the simplest geometrical objects that a system can handle*. Primitive surfaces are a subset of simple parametric and/or implicit surfaces. However, since they are based on common shapes, they are described in a more semantic way. Hence primitives facilitate the comprehension of complex shapes. For example, a cylinder is defined by its axis, its height and radius, and an axis-aligned cube is defined as a centre point and a width.

Discussion

As a particular subset of surfaces, primitives can always be expressed as an implicit or parametric model. In addition of sharing the advantages of these models, the simplicity of primitive surfaces minimises the models limitations and

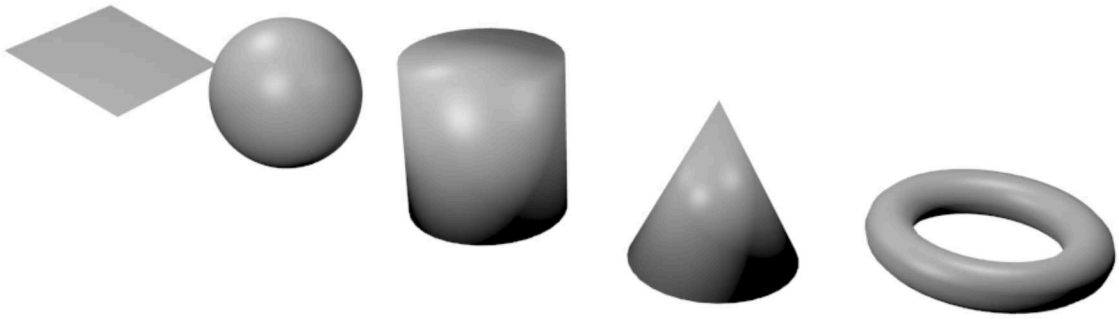


Figure II.12. – The set of important primitives: plane, sphere, cylinder, cone and torus.

point cloud approximation with primitives can be achieved efficiently. Another advantage of primitives is that it is possible to constrain the domain of definition of primitive models (e.g. only model a truncated cone, or a partial torus). Also, the computation of differential geometry is straight forward and accessible at any point on the surface. The fixed set of primitive surface model may constitutes a shortcoming depending on the context of application. Alike elevation surface models, primitives do not offer any local control on the surface. The rigidity of these models does not provide the ability to model general complex objects. But it offers a great advantage when facing important occlusions since it naturally extrapolates the geometry in missing data based on available information. Also, considering that complex objects often can be approximated with a more simple appropriate collection of primitives, this choice of abstraction in geometric modelling reveals benefits (e.g. the body and the engines of a plane can be approximated by a cylinder, the wings as planes, and the cockpit as a cone).

Conclusion

Primitives are often used when assumptions are made on the point cloud to approximate. As an abstraction of daily objects they provide a clear understanding of these shapes. They also gives direct access to important human-readable measurements. With their extrapolation of missing data, they represent a convenient and intuitive model of surface for the decomposition of an object into several parts. They are mainly used for shape extraction, reverse engineering analyses and measurements, even on complex shapes that can be approximated by simpler surfaces. Hence, instead of modelling a point cloud with a general surface model, one can analyse the accuracy of a point cloud approximation with different primitives in order to segment and classify the data [113, 181] or detect particular primitives of interest in the data [27, 161].

2.4. Conclusion

Now that the strengths and weaknesses of the available surface models have been highlighted, it is possible to identify the best appropriate model for each application. Our goal is to extract tubular shapes (surfaces of revolution around an axis) while minimising the computing time. Finally, we intend to apply our methods to large, dense, noisy and occluded point clouds with varying point density. This consideration along with our objectives motivated our choices of surface models.

Even if mesh models provide convenient control over restored surfaces, meshing dense point clouds is time consuming and would either entail high resource requirements or complex simplification issues. Also, we pointed out earlier that mesh models are sensitive to noise and even more to large occlusions. Hence we did not consider mesh models to be an appropriate solution.

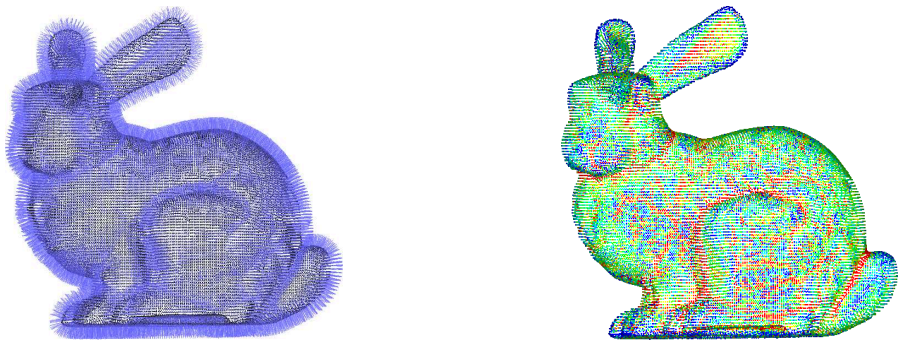
Using B-Splines and NURBS models for surface approximation raises points parametrization issues in dense unstructured point clouds and requires a complex estimate of knots and weights. In addition overcoming occlusions with these models is problematic and incomplete data reconstruction represent a challenge. Therefore B-Spline and NURBS do not meet our criteria for robust shape extraction from highly occluded point clouds.

Implicit models can handle surfaces of revolution but, even though they are powerful for surface reconstruction, they are less adapted to segmentation and pattern recognition problems. Therefore implicit models appeared to be inappropriate for our objectives as also they tend to face limitations in presence of occlusion and incomplete data.

Elevation surface models provide efficient surface approximations using least square optimisation. This computational efficiency along with the direct access to points on the surface and to the differential geometry makes elevation surface models attractive for rapid algorithms. Hence, even though they require the computation of a local basis, we chose to use elevation surfaces for our fast normal estimation method.

We chose to use parametric models and primitives for our tubular shape extraction method. Like implicit models, parametric models can easily handle surfaces of revolution around a given central curve, yet, they require less computations. With an assumption about the content of the data, primitive models are adapted to guide the surface approximation. Moreover, as pointed out, they rigidity allows them to model full shapes from occluded and incomplete data.

3. Elements of geometry on point clouds



(a) (b)

Figure II.13. – Estimation of normal vectors and mean curvature on the Stanford bunny model.

Point clouds greatly suffer from the lack of geometric information because they only consist in point locations. Data points are sampled on surfaces, hence many efforts have been put to develop methods that approximate the local geometry of the underlying surfaces directly from the point cloud. Differential geometry is a fundamental tool for studying smooth surfaces, and this is even more important on point cloud where there is a need to overcome the deficit of structural information. Indeed, the estimation of first and second order geometric attributes are indispensable for the maturation of numerous applications: feature detection [70], segmentation [176, 198], surface reconstruction [81, 93], point cloud modelling [113, 181], and many others. In particular, normal vectors, curvatures and principal curvature are major elements of geometry. Existing works focus on the approximation of such elements on point clouds. Most of them are based on local computation of differential geometry. However, even though the notion of "local analysis" is well defined on meshes and organised point clouds, this notion has to be adapted to unorganized point clouds.

In order to better apprehend the computation of geometric elements on point clouds we first present the principal definitions of neighbourhood used for local analyses on point clouds. Then we introduce the data structures used to efficiently compute such neighbourhoods before exposing the notion of normal vector and curvature on point clouds.

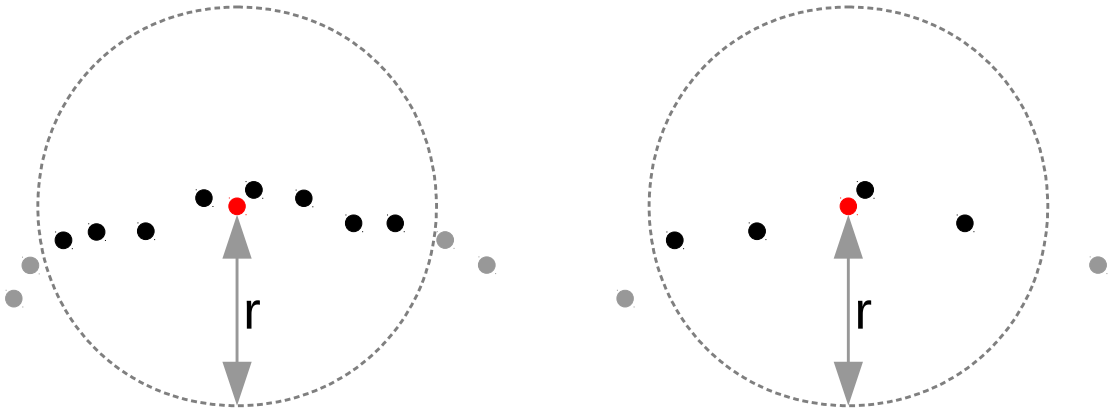


Figure II.14. – The spherical neighbours of a query point p in red are the black points inside a ball of radius r centred at p. The sampling density influences the number of neighbours.

3.1. Neighbourhood definitions for local analyses

Point clouds are discrete samples of continuous surfaces. This is why, inspired from other discrete representation of a continuous object (e.g. images or meshes), local analyses on point clouds are based upon the notion of neighbourhood around a given point. In addition to differential geometry analyses, the definition of a local neighbourhood has many applications in clustering, segmentation or feature estimation. However, the lack of connection between the data points cannot lead to an evident definition of neighbourhood on raw point clouds. Hence different solutions have been proposed to address this issue and extract a pertinent neighbourhood of a given point. Spherical neighbourhood and k-neighbourhood are the most used among other proposal.

3.1.1. Spherical neighbourhood

Spherical neighbourhood have been introduced as a simple and intuitive definition of a neighbourhood on a point cloud. A spherical neighbourhood is defined according to the Euclidean distance (Fig. II.14). The set of neighbours N_p of a point $p = (x, y, z)^t$ is defined as the set of points contained inside a sphere of radius r centred on p: $N_p = \{p' \mid \|p - p'\| \leqslant r\}$. This definition offers several advantages besides being intuitive. Since the radius of the sphere r is set by the user, it allows a large flexibility on the definition of the size of the neighbourhood to be considered. In addition, multi-scale analyses are easily accessible and ef-

(a) (b)

Figure II.15. – The k-neighbours ($k = 5$) of a query point in red are shown in black. The point cloud density influences greatly the distance from a point to its k-neighbours.

fortless to implement [81]. Spherical neighbourhood ensures the proximity of the corresponding neighbours and can be computed efficiently with the help of octrees (see Section 3.2). These advantages justify the choice of spherical neighbourhood in a wide range of applications including, but not limited to, features computation, clustering and classification.

However, this notion of local neighbourhood also includes undesired elements. Since it is only based on the Euclidean distance separating the points, the number of neighbours n of a point p is not known in advance. Even more, this number may differ for each point and varies according to the sampled surfaces and the point cloud density. Whereas algorithms can benefit from the flexibility of the radius search r, this parameter has to be set carefully in order to obtain a pertinent neighbourhood. It requires a great expertise to select this radius according to the size of the details to capture and the point cloud density. This is even more true when the sampling density varies along the data set. Finally, this definition of a neighbourhood is not invariant to point cloud scaling. i.e. if P' is a scaled version of a point cloud P, the set of neighbours of a point p will differ from the set of neighbours of its corresponding point p'.

3.1.2. k-neighbourhood

A second approach for defining the local neighbours of a point concurrences the spherical neighbourhood search: the k-neighbours N_p of a point p are defined as the k points closest to p (Fig. II.15):

$$N_p = \{q_i, i = 1, \cdots, k \mid d(p, q_j) > d(p, q_i) \; \forall \; j\} \tag{II.12}$$

With this definition, the negative effects of the spherical neighbourhood va-

nishes but others appear. Indeed, the parameter k fixed by the user guarantees a constant number of neighbours for each point of the data, and makes the definition of neighbourhood invariant to point cloud scaling. k-neighbourhood can achieves better results on densely sampled parts since it is able to provide a sufficient number of points in a smaller area around the query point. Moreover, in presence of outliers, the k-neighbourhood is assumed to be more robust [137] because the distance from the query point to the outliers is usually greater. Similar to the spherical neighbourhood, k-neighbourhood is widely used in various processing including feature detection [4], segmentation [52], skeletonization [208], or surface approximation and simplification [4, 138].

However, the k-neighbourhood does not ensure that the neighbour points are located in a radius around the query point. Indeed, depending on the data, it may contain some far away neighbours or outliers (Fig. II.15). Also, the computation of the k-neighbours might be less efficient than that of the spherical neighbourhood, even using a KD-Tree as an accelerating structure. This is why several algorithms have been developed to produce efficiently an approximation of this neighbourhood called Approximate Nearest Neighbours implemented for example in the FLANN library.

3.1.3. Other neighbourhood definitions

Spherical neighbourhood and k-neighbourhood both present undesired effects. Several other definitions have been proposed in order to minimise the limitations encountered but are not wide spread hence we do not provide an extensive insight of these variants. A simple solution is to fuse both definitions and only consider the k-neighbours inside a radius ball. A more appropriate option could be to use cylindrical neighbourhoods [184]. Given a point p and its normal n, the cylindrical neighbourhood includes all the points inside a cylinder of axis n, fixed height h and fixed radius r. Finally, since the sampling density influences greatly both the selection of the radius for a spherical neighbourhood and the choice of the value of k in the k-neighbourhood, some other solutions have been proposed based on multi level analyses [81], or on adaptative cylindrical neighbourhood [105].

3.1.4. Conclusion

As the basic element of a large field of processing, the notion of neighbourhood on point clouds constitutes a crucial element of study. It replaces the

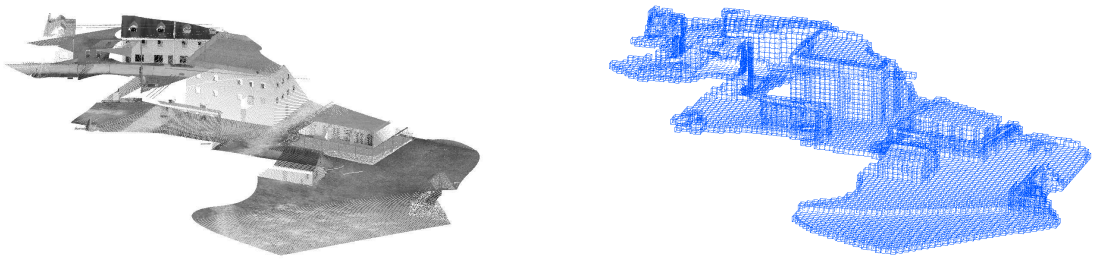
well known neighbourhood operators available on meshes, images and organized point clouds. It is essential for local analyses of the point cloud, and in particular for estimating the geometry of scanned surfaces. Even though several definition of neighbourhood have been proposed, to this day the spherical neighbourhood and the k-neighbourhood remain the most popular definition.

The neighbourhood definition on point cloud tackles the lack of structural information in the data sets. It allows to develop local analyses, to derive the pendant of edges or pixel neighbourhood and to discretize the notions of differential geometry on point clouds. However, the computation of point neighbourhoods can be time consuming with brute force methods. Hence several data structures have been employed on point cloud to accelerate these computations.

3.2. Data structures

Unorganized point cloud are difficult to handle in an efficient manner because of the absence structural information. In addition the tens of millions (or more) points acquired by t-LiDAR instruments represent an obstacle for computationally efficient processing, besides the challenges of adapted visualisation and storage. Point cloud processing algorithm have to take into account the large amount of data to offer viable solutions to existing problems: an algorithm with accurate results is not attractive if it cannot be ran on real data sets. Developing fast algorithms is hence a challenging task. In particular, previously introduced points neighbourhood are crucial for differential geometry analyses and for a large set of further algorithms. Consequently there is a strong need of using adequate data structures with point clouds in order to accelerate the various algorithms and to facilitate both the manipulation of and the visualisation of point clouds.

These structures are often based on space partitioning and thus provide major utility in Euclidean clustering or neighbourhood search (e.g. FLANN library, PCL library). In addition, they provide another representation of point clouds that can be used to apply methods derived from the volumetric domain. For these reasons, data structures have become indispensable for an efficient point cloud processing. The common data structure applied on point cloud can be divided into two major families that are regular grids and space partitioning trees.



(a) (b)

Figure II.16. – The initial point cloud (a) is structured into a regular 3D grid (b).

3.2.1. Regular grids

Regular grids are the simplest data structure available on point clouds. They transform a point cloud into a volumetric image that can be efficiently processed with existing algorithms. Among many other applications, the regular are used to accelerate the computation of points' neighbours in constant time [127]. They can also be adopted to perform point cloud segmentation via connected component detection, or simplification via regular downsampling (see Section 4.3). Finally, combined with implicit functions (e.g. signed distance function) and a marching cubes algorithm, they provide a convenient tool for implicit surface extraction.

Definition and content

A regular grid is constructed by dividing the 3D Cartesian space into equal-sized cubes also referred to as voxels (Fig. II.16). Grids can be seen as enhanced discrete representations of a point cloud since each voxel keeps track of the points it contains. In addition, voxels can store a value to produce a greyscale volumetric image. This value may reflect the presence and absence of points in a binary manner, or the local density of the point cloud by considering number of points inside each voxel. Voxels can also store more elaborated values such as variance [52] or Plant Area Indices (PAI) [12, 55, 173].

Regular grids are common structures in point cloud processing because they offer a direct link between points to voxels. This simple structure offers a regular subdivision of the point cloud from which points' neighbours computation can be accelerated. When exploited to produce a volumetric image, it gives access to a series of established algorithms including filters, or morphological analyses. Voxel neighbourhood is straightforward, hence discrete computations such as gradients are efficient. However, the choice of the size of the voxels remains a limitation since it has to be set carefully according to the data set for each ap-

plication. Furthermore, the regular subdivision scheme might be inappropriate for certain utilization since neither the local density nor the local geometry of the data is taken into account to produce the final structure. Despite these drawbacks, regular grids remain one of the most frequently used structure in point cloud processing because in addition to offer efficient computation of points neighbourhood, they introduce 3D image processing algorithms into point cloud analyses.

3.2.2. Space partitioning trees

Aside from regular grids, a major set of data structures handle point clouds by dividing the 3D space into potentially non regular elements: space partitioning trees includes several structures that provides a flexible construction of the space division. They offer additional qualities compared to regular grids and are part of the solution to many problems.

General definition

Space partitioning trees greatly differ from regular grids: they are the result of a recursive subdivision of the space and cannot be stored as images (i.e. array of elements). Instead, the recursive subdivision leads to a tree structure where each node represent a portion of the space. Starting from a tree node defined as a boundary volume of the point cloud, each node of the tree will be recursively split by hyperplanes according to one or more constructive rules. The choice of the cutting planes during the subdivision and the cutting order varies among the different space partitioning trees. Even though many variant of the space partitioning trees exist (some of which do not rely on cutting planes), the most commonly used in point cloud processing are octrees and kd-trees which entail advantageous node properties.

Octrees

Octrees are a major element of space partitioning structures. They are the 3D equivalent of the 2D quadtrees: each node is subdivided into eight equal-sized octant derived from three cutting planes parallel to the 3D space axes (Fig. II.17). The classical octrees subdivision is guided by the point density of each node: the constructive rule stops the recursive subdivision when a node contains less than a predefined fixed number of points. Some supplementary rules are often introduced to regulate the subdivision. For example, it is possible to impose a minimum size for the octree nodes in order to ensure an appropriate scale for further analysis. Setting a maximum depth for the tree may also be desirable

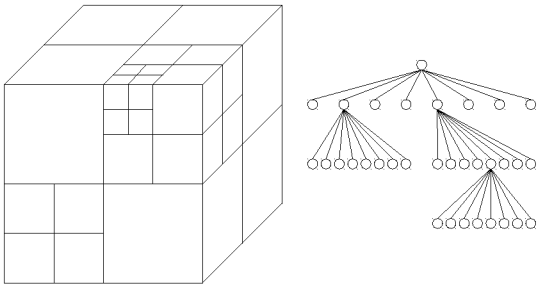


Figure II.17. – The recursive subdivision of octrees into octants is represented as a tree structure.

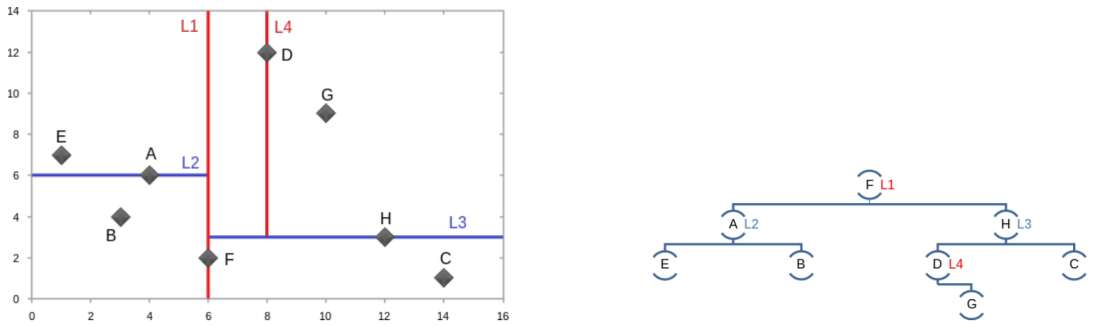
since it determines the maximum cost of tree traversal operations.

With the subdivision into octant, octrees offer a powerful way to compute spherical neighbourhoods (see [58] for a comparison of neighbour finding approaches). Ray tracing algorithms have also been adapted to octrees as a solution for fast intersection problems [177]. Depending on the application, different implementations of octree structures have been proposed. Some of them specially focus on memory efficiency and provide an efficient solution for handling large data sets, up to several billions of points [57].

With their simplicity and flexibility, octrees are appealing data structures that are easy to use. Even though they are mainly exploited for accelerating spatial neighbourhood searches, they play a major role in diverse complex processing and are widely spread in point cloud processing and visualisation [93, 170, 201].

KD-trees

Depending on the situation, the subdivision into equal size cubical octant may not be the most appropriate. When there is a need to guaranty the balance of the space partitioning tree, it is preferable to use KD-trees. Contrary to octrees, KD-trees are binary trees (i.e. each node only have two children). At each level of the recursive subdivision, instead of splitting a node into eight octant, the KD-tree only generates two children separated by an axis-aligned plane (Fig. II.18). The choice of the axis selected for a node cut is cyclic (e.g. x-aligned cut, then y-aligned cut followed by z-aligned cut, then x-aligned cut again and so on). However, the split do not divide a node in half. Instead, the planes typically cuts a node at the median point along its direction (Fig. II.18). This choice ensures a balanced tree and establishes useful structure for k-neighbourhood searches.



(a) (b)

Figure II.18. – The KD-tree commonly subdivides a given space using a medial cutting hyperplane (a). This recursive subdivision generates a binary tree (b).

The KD-trees concurrences the octrees in numerous domains. Indeed, the construction of KD-trees ensures that the created tree is balanced, and hence automatically bounds the maximal depth of the tree. The constructive rules also guaranties better properties of the subdivision regarding to the number of points in each node. The KD-trees have proven to be more adapted to computes k-neighbourhoods and often outperform octrees for spherical neighbourhood searches [58]. However, sometimes, the cubical subdivision is preferable. Also,note that inserting points in a KD-tree is expensive as it requires costly operations to rebalance the tree.

Despite their advantages, KD-trees have not overtaken octrees. Even though they appear to be the most suited data structure for neighbourhood search operations, the lack of immediate relation between nodes restricts their usage. Whereas octrees are part of many algorithms, KD-trees applications are often limited to spatial neighbourhood queries.

Discussion and conclusion on SP trees

The family of space partitioning trees present some major advantages to handle point clouds efficiently and provides a set of fast neighbourhood searches algorithms. Among the variety of available structures (variants of KD-trees, Ball-trees, etc), octrees and KD-trees, with their respective strengths, have proven to be the most appropriate for managing point clouds. Hence, they are found in almost every procedure to accelerate the computations. In chapter III, we propose a solution that takes advantage of the octrees qualities and speeds up the normal estimation procedure: with some new subdivision rules for the space subdivision, we introduce a derived version of the octrees named *anisotropic octrees*.

3.3. Elements of differential geometry

The approximation of the local geometry is one of the most important preconditions of point cloud analysis. This includes the estimation of first and second order geometry (normal vectors and curvatures) of the point cloud. Under the general assumption that point clouds are sampled on smooth surfaces, normal vectors and curvature are often derived from differential geometry.

Many algorithms require an estimate of normal vectors and curvature on a point cloud in order to properly extract high level information from a point cloud. For example, they constitute the main element to achieve point cloud skeletonization [96, 191], segmentation [176], shape extraction [181] or surface reconstruction [81, 93]. In particular, normal vectors are a prerequisite for our tubular shape extraction method. Therefore in this section we will introduce both recall the notions of differential geometry on surfaces and introduce the state-of-art related to normal vector estimation on point clouds. These elements also motivate our work on *anisotropic octrees* and fast normal estimation detailed in chapter III.

3.3.1. Normal vectors

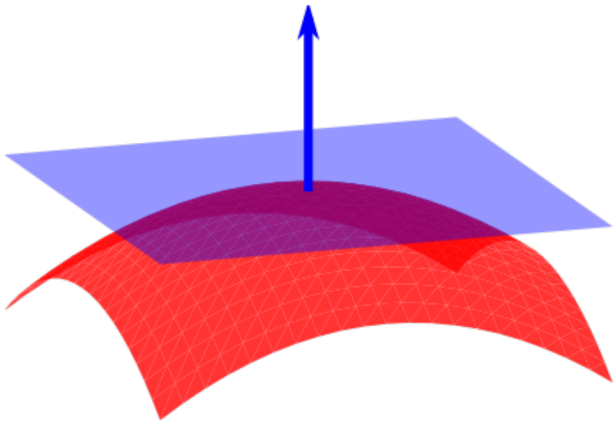


Figure II.19. – The normal vector at a point p on a surface is the vector orthogonal to the tangent plane at p pointing outward the surface. Adapted from Wikipedia.

The study of local geometry of smooth surfaces is related to differential calculus applied to 2D manifolds. Considering a smooth surface S, the normal at a point $p \in S$ is a vector orthogonal to S at p. Hence, it is the direction orthogonal to the tangent plane at p (Figure II.13a). Normal vectors also provide an

orientation of the surface (they points outward the surface). For a more complete description of the local geometry of a surface the reader is referred to [16]. Normal estimation on point cloud is greatly inspired from knowledge on smooth surfaces because it is assumed that points are sampled on such surfaces. Hence we now recall some elementary notions for smooth surfaces before introducing the approaches available for estimating normals on point clouds.

Normal vectors on parametric surface models

Given a smooth regular parametric surface model $M(u, v)$, the tangent vectors t_u and t_v obtained by the partial derivates of M define the tangent plane at a point $p(u, v)$. The normal vector n_p at $p(u, v)$ is orthogonal to t_u and t_v and is given by:

$$n_p = t_u \wedge t_v \tag{II.13}$$

where $\wedge$ denotes the outer product.

Normal vectors on implicit surface models

On implicit surface model defined as the zero level-set of a smooth implicit function $F(x, y, z)$, the gradient of F at a regular point p (the gradient of F at p is non zero) is the normal vector n_p at p:

$$n_p = \nabla F(x, y, z) \tag{II.14}$$

Normal vectors on mesh models

Because meshes consist in faces, they are piecewise linear models and the resulting surface is continuous but not smooth (they are neither $mathcalG^1$ nor differentiable). However, several expressions have been developed to estimate normal vectors on mesh models that aims at approximating the "differential" normals when the model converges towards a smooth surface [134]. The most widespread approximates the normal n_v of a vertex v as the weighted average faces normals in a one-ring neighbourhood around v:

$$n_v = \sum_{i=0}^{k} w_i n_i \tag{II.15}$$

where n_i is the normal vector of the i^{th} face in the one-ring neighbourhood of v containing k faces, and the weights w_i can be derived from several properties of the faces including angles, area.

Normal vectors on point cloud

The computation of normal vectors on point cloud is challenging since the sampled surface is unknown. Nevertheless, this problem has been widely studied (see review by [99]). In the following we present some existing normal estimation methods in order to highlight their computational cost that constituted a limitation to our researches and motivated our development of a fast algorithm. Normals estimation can be divided into four major approaches: *optimisation-based*, *local-weighting-based*, *Hough transform*, and *Voronoï-based*.

The *optimisation-based* approach locally approximate the point cloud with a predefined surface model (often parametric) from which normals can be computed. The two main models fitted to the data are planes [81], and quadratic surfaces [99]. However other second-order surfaces such as spheres have also been used to approximate the point cloud [217]. Planes can be fitted by using either classical least square fitting, or principal component analysis (PCA) [70]. In *local-weighting-based* approach, normal directions are estimated by different combinations of the normals deduced from a point and its neighbours [89]. This approach is similar to normal computation on meshes and can be either applied on a local Delaunay triangulation of the point cloud, on kNN graphs, or on Reinmann graphs [99]. *Hough transform* is an alternate approach that considers a point and its neighbourhood. For each point of the cloud a reduced amount of triplets is selected, the Hough transform then provide a reliable accumulator from which robust normal directions can be extracted [29]. This point-wise approach can be supported by a neuronal network to improve its robustness to noise and outliers [30]. The *Voronoï-based* approach considers the shape of each Voronoï cells [6]. The algorithms use Voronoï poles to estimate normal directions caan also provide additional information on curvature [50, 132].

Conclusion

With no information about the underlying surface normal estimation on point clouds is a challenging task. Existing approaches often rely on local approximation of point clouds with surface models, and normals are estimated using differential geometry or mesh-inspired approximations. On large point clouds, the optimization-based approach is preponderant since other approaches tend to be more time consuming. However, even optimization-based methods face limitation when confronted to data sets containing up to several tens of millions of points. Inspired by this approach, we address this issue in chapter III of this

thesis and propose an fast alternative approach to estimate a normal vector field on a point cloud.

3.3.2. Curvature

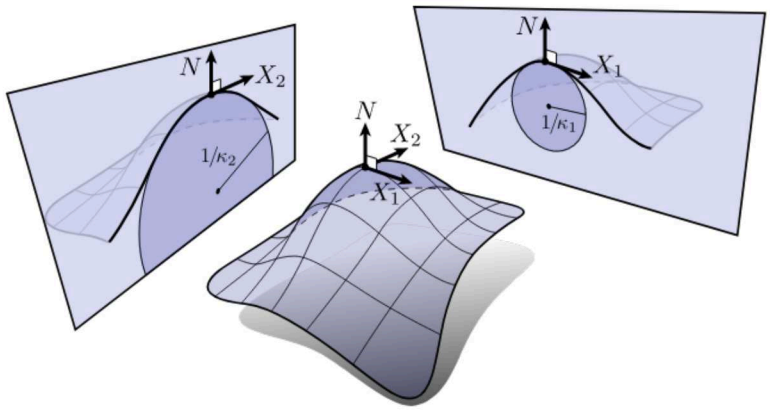


Figure II.20. – Principal directions are the orthogonal basis vectors in the local tangent plane in which the curvature matrix is diagonal. The notion of curvature can be derived from the radius of an osculating circle. Adapted from [182].

While normal vectors are major elements to be considered, they might be insufficient for some applications. Deeper analyses and point cloud processing (including feature detection [90], point cloud simplification [110]) often require an estimate of the curvature value or principal curvatures. Curvature estimation has been widely studied on surfaces and meshes for a wide range of applications [101, 106, 175]. Our normal estimation method detailed in chapter III also provides an curvature indicator, even though this notion is not central. Therefore, we now briefly recall the definition of curvature necessary to apprehend this work.

Definition

Roughly speaking, the curvature of a surface is an indicator of "the local variation of normal vectors". The notion of curvature is well defined in differential geometry, but intuitively, the notion of curvature can be considered in different ways. Among them, the curvature along a direction $\overrightarrow{d}$ of the tangent plane can be seen as the inverse of the radius of the osculating circle of a curve drawn on the surface along $\overrightarrow{d}$ (Fig. II.20).

More precisely, consider a smooth surface S and a point $p \in S$. The partial

derivatives S_u and S_v of S at p provide two tangent vectors. Let us denote by n the unit normal at p, thus given by $n = \frac{S_u \wedge S_v}{||S_u \wedge S_v||}$. Then in the basis of the tangent plane (p,S_u, S_v), the curvature is a quadratic form K_p which matrix is given by:

$$K_p = I_p^{-1} II_p \tag{II.16}$$

where:

$$I = \begin{bmatrix} < S_u, S_u > & < S_u, S_v > \\ < S_u, S_v > & < S_v, S_v > \end{bmatrix} \tag{II.17}$$

($<,>$ denotes the dot product) is called the first fundamental form and:

$$II = \begin{bmatrix} < \frac{\partial^2 S(u,v)}{\partial u^2}, n > & < \frac{\partial^2 S(u,v)}{\partial u \partial v}, n > \\ < \frac{\partial^2 S(u,v)}{\partial u \partial v}, n > & < \frac{\partial^2 S(u,v)}{\partial v^2}, n > \end{bmatrix} \tag{II.18}$$

is the second fundamental form. The matrix K_p is symmetric. Therefore it can be diagonalized in an orthonormal basis of eigenvectors. Let $(\overrightarrow{d_1}, \overrightarrow{d_2})$ denote this orthonormal basis and let k_1 and k_2 be the associated eigenvalues. The vectors $\overrightarrow{d_1}$ and $\overrightarrow{d_2}$ are called the principal curvature whereas k_1 and k_2 are the minimum and maximum curvatures. From these elements the Gaussian curvature of S at p is then obtained as:

$$K = k_1 k_2 \tag{II.19}$$

and the mean curvature is computed as:

$$H = \frac{k_1 + k_2}{2} \tag{II.20}$$

Like normal vectors, curvatures and principal curvatures can be estimated on point clouds via *optimization-based* approach: first the point cloud is locally approximated by a surface, then curvatures are computed directly on the surface and are assigned back to the data points.

Unfortunately, in spite of the efforts made to define curvature estimates on point clouds, they do not seem to achieve reliability. However, this geometrical measure offers a better apprehension of the scanned surfaces. Estimating curvatures on point clouds also allows to benefit from a large collection of algorithms developed for surfaces.

4. Point cloud preprocessing

It is a common practice to apply pre-processing steps to t-LiDAR acquisitions in order to enhance the data quality. It is therefore important to identify these treatments and their potential impact on point clouds. Hence we now introduce some of the major pre-processing methods applied to t-LiDAR point clouds and their principles.

In an attempt to minimise the acquisition artefacts, numerous preprocessing have been proposed to tackle the t-LiDAR acquisition limitations and enhance the results of further algorithms. The non-homogeneity of the sampling density being generally addressed by the selection of adapted neighbourhood definition, the preprocessing steps focus on the three other problems. First, occlusions are minimised via multiple scan acquisitions and point clouds *registration*. Second, *denoising* and *filtering* methods have been proposed to detect and remove noise points from the raw data. And finally, point cloud *simplification* are set to reduce the size of the data while preserving the important geometrical elements.

4.1. Registration

It is a common practice in t-LiDAR scanning to acquire the same environment from multiple points of view. This tends to minimise occlusions since each acquisition captures a different parts of the environment (Fig. II.21). Point clouds from multiple acquisitions then have to be aligned together in a single data set. This so called registration problem aims at estimating the transformation existing between two or more data sets. In a generic case, this transformation can be non-rigid (i.e. including constant or varying scaling, reflections, etc). However, remote sensing acquisitions of point clouds benefit from a great accuracy of the sensors, and acquired scenes are considered static. Hence the problem of point clouds registration only focuses on estimating a rigid transformation T, from a source data set S to a reference data set R, composed of a translation t and a rotation r:

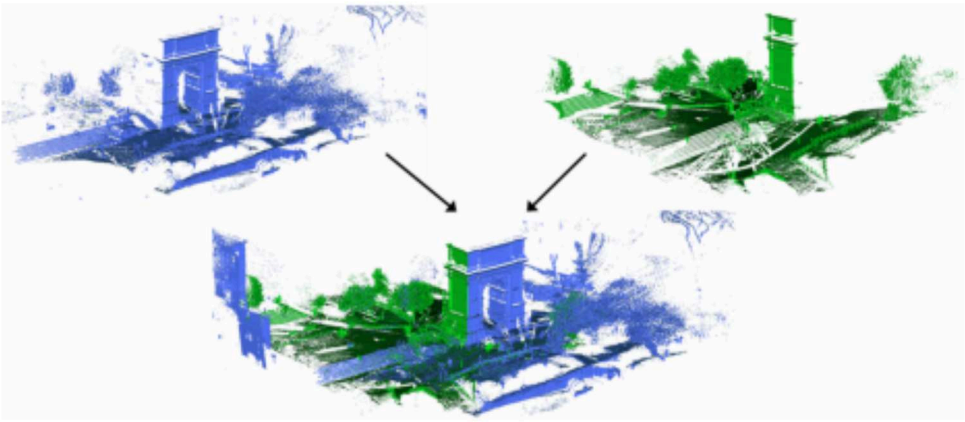


Figure II.21. – Two t-LiDAR acquisitions from different positions are registered to generate a new point cloud minimising occlusion.

$$S = TR = rR + t \tag{II.21}$$

where r is a rotation matrix and t a translation vector.

Numerous solutions have been presented in the literature for 3D point clouds rigid registration (for example see survey [14]). The main steps of a registration algorithm are: (1) select a set of n *anchors* in a source cloud and a set of n *anchors* in the target cloud, (2) find a pair-wise matching of the *anchors*, and (3) estimate the transformation T providing the best solution for the matching. Registration algorithms can be divided in two approaches depending on the nature of the *anchors*. In the latter we denote as *point-based* the approaches considering data points for *anchor* candidates, and *feature-based* the approaches considering enriched points or higher level objects as *anchors*.

4.1.1. Point-based approaches

In point based-approaches, the estimation of the transformation T is based on a point-to-point correspondence from the source to the target cloud: when this correspondence is known (e.g. by user interaction), T is defined as the transformation that minimises the euclidean distance between the pairs of points:

$$T = argmin \sum_{1}^{n} \|(Rs_i + t) - r_i\|^2 \tag{II.22}$$

This problem can be solved using a Singular Value Decomposition (SVD) of the cross-correlation matrix $C = sr^t = USV^t$ [14]. However, this solution is limited since it requires a precise estimate of the points correspondence which is non trivial. The Iterative Closest Point algorithm (ICP) was introduced to address this issue [20, 216]. This algorithm avoids the need of *a priori* knowledge of points correspondence by solving the problem iteratively. At each iteration the information of closest neighbour is used as a heuristic to determine point correspondence. This information is combined with the SVD decomposition to obtain the transformation T. This procedure is then repeated until the algorithm converges. Several improvements have been proposed to improve the convergence rate and enhance the accuracy of ICP [69, 127]. Additional variations of the definition of the transformation T have been presented where T minimises a point-to-surface distance rather than a point-to-point distance [123]. Along with its modified versions, the ICP algorithm is currently the reference for point cloud registration because it is intuitive, simple to implement and provides accurate results [14].

4.1.2. Feature-based approaches

In order to tackle negative effects of noise and outliers on the matching process, and to reduce its computational time, feature-based algorithms use higher level information as *anchors*. Numerous methodologies introduced key points such as Scale-Invariant Feature Transform (SIFT) descriptors or corners [26], skeletons [218], and specific patterns such as planes [210] which prove to improve the robustness and convergence rate of the algorithms. In the context of forest point clouds, detected trees can be used as anchors to compute a coarse registration [23]. These solutions tends to be less sensitive to noise since they take into account either confident anchor points or larger reconstructed objects less affected by outliers. However they depend on the robustness of the feature extraction algorithms. In addition, only a small set of anchors is considered during the optimisation procedure, hence these methods perform usually faster than traditional point-based algorithms.

4.1.3. Other approaches

Coarse point cloud registrations can also be achieved from various information. For example, the intensity of the points (reflectance of the materials) can also be exploited [26]. As a statistical tool, PCA have also been employed to achieve fast but coarse registration by aligning the principal axes [14]. Those

Figure II.22. – T-LiDAR acquisition suffer from noise and outliers.

approaches are provided for efficiency but faces numerous limitations and therefore are only used as preprocessing steps for fine registration algorithms. The fine solution chosen in forestry is a semi automatic procedure involving the identification of spheres in the point cloud that have been positioned in the field before acquisition (see the FaroScene software documentation).

4.1.4. Conclusion

Even though registration algorithms can be advantageous in computing time, convergence rate and accuracy, they still faces some limitations with noisy t-LiDAR acquisitions besides artefacts caused by the wind in forest environments. Moreover, the co-registration of multiple scan has to be done carefully in a global manner to avoid the error accumulation. Nevertheless multi-scan acquisition is a common usage that reduces occlusion and captures more details that facilitate reconstruction procedures and other geometrical analyses. In counterpart it raises additional difficulties because it increases the non-homogeneity of the sampling density and generates huge data sets. As stated before, our work focuses on raw data sets, hence this step is optional for our methods that are able to handle incomplete point clouds.

4.2. Outliers detection and noise reduction

Point clouds acquired from remote sensing technologies often contain noise and outliers that affect the quality of geometry estimation (Fig. II.22). Numerous solutions have been presented to attenuate the effect of these points on the

final results (see review [75]). The data set we processed along our work were severely affected by noise and outliers. Therefore our developments aimed to be robust and it is important to apprehend the different ways of handling such limitations. These solutions can be divided into two main classes: *denoising* approaches (the term "*thinning*" is sometimes used in the literature) tend to modify the point clouds by moving the points onto the approximated real surface, whereas *filtering* approaches identify outliers and noise points before removing them from the data.

4.2.1. Denoising

Denoising approaches on point clouds have often been inspired by image processing methods. For example, the definition of the Laplacian filter and the heat equation frequently used on images have been adapted to point clouds in order to reduce the noise and outliers by smoothing the point cloud [158, 214]. Spectral analyses such as the powerful Discrete Fourier transform have been proposed to reduce noise, and finds other applications [155]. The bilateral filter was successfully adapted on a point cloud to thin the surface thickness due to noise [64]. Robust PCA proposed in [107] provides a local first order approximation (plane) of the underlying noise-free surface. Points are then projected on the locally estimated surface to reduce the noise. Similarly, numerous denoising methods are based on adaptations of the moving least squares (MLS) approximation [4, 49]. However, the above solutions result in an isotropic smoothing which does not preserve the features of the point clouds.

4.2.2. Filtering

Point cloud denoising reduces the effect of noise and outliers but moves the data points. This modification of the geometry is in some cases undesirable. An other approach for handling noise is to filter outliers and noise points without affecting the rest of the original data set. Considering a locally regular point spacing, many filters use statistical methods to detect outliers and noise [189]. The mean distance from a point to its k-neighbours has proven to be a reliable variable for the identification of outliers [175]. But other metrics are present in the literature [214] and distance from a point to an ideal surface may also be considered discriminant. Extension of the 2D convolution filters on images have been proposed in the CompuTree software and applied with success on a voxelisation of point clouds to diminish the amount of outliers and filter undesirable elements from an a priori of its structure. [202] used a majority voting scheme in

an iterative refitting procedure to obtain a robust discrimination of outliers from a set of *regular* and a set of *irregular* points. This way they spatially separate data points from outliers clusters and isolated outliers, which are more easily filtered while preserving the noise-free points. Finally, to correct noise induced by the acquisition techniques, several filters based on signal intensity are present in proprietary software and commonly applied to the point clouds (on top of built-in filters).

4.2.3. Conclusion

In the context of precise measurements from t-LiDAR acquisitions, denoising methods are to be avoided: the sensor's accuracy is considered sufficient enough to provide an accurate sampling of the points on the surface. Smoothing the point cloud would alter the reconstruction of the actual surface and introduce errors in the resulting measurements. Therefore, these noise reduction methods are not appropriate to deal with the constraints of this thesis. Filters that do not modify the data are thus preferred, but the obtention of a noise-free point cloud requires a delicate parametrization of the algorithms. Overall, in the context of this thesis, such pre-processing are optional since our goal is to produce reconstruction methodologies robust to these limitations. Hence even though a lesser amount of noise generates better quality results (see chapter IV and V), in the general case no filter is applied on our data.

4.3. Simplification

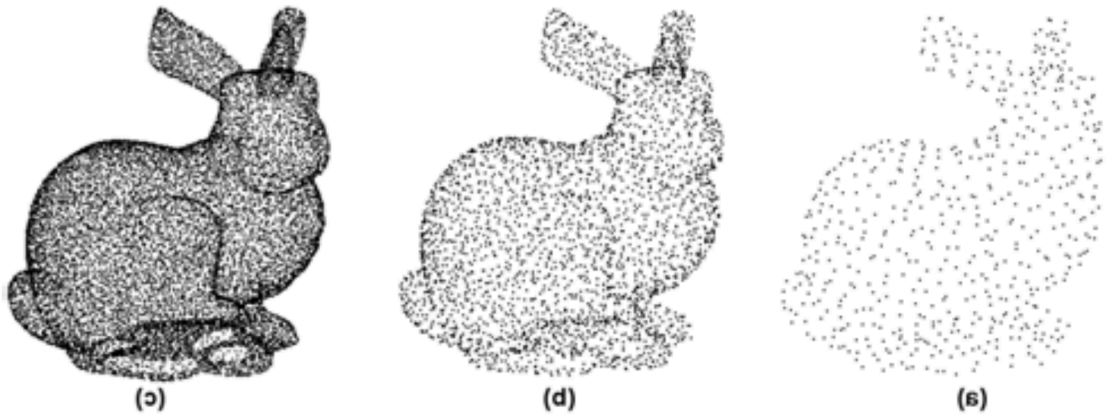


Figure II.23. – Simplification of the Stanford bunny point cloud representation. Adapted from [138]

With the great accuracy of point clouds acquisition techniques comes the problem of data size. Indeed, point clouds acquisition often generate extreme amounts of points which leads to data redundancy and makes the data processing computationally costly. In order to reduce the computational time of point cloud processing, point cloud simplification algorithms have been introduced. During our thesis we compared our developments to other algorithms that required the point cloud to be simplified first (see chapter V). Therefore, we now present an overview of point cloud simplification methods while additional detailed can be found in the given references.

The simplification of a point cloud P of n points is the procedure of producing a point cloud Q with m points, $m < n$. The points of Q can be a subset of P or not. The simplification problem can be defined in several ways [187]: either the number of points m in the resulting point cloud is specified, or a maximum error between the input and the simplified point cloud can be stipulated. Greatly inspired by the numerous existing studies on mesh simplification [156], point cloud simplification algorithms can be divided into three major approaches: iterative simplification, clustering and particle simplification. Note that these approaches can be either global (i.e. the sampling density of the simplified point cloud is constant) or adaptative (i.e. the feature of the input point cloud are preserved in the simplified version thanks to a higher point density).

4.3.1. Iterative simplification

Inspired by the point removal and edge contraction approaches on mesh models, point cloud simplification can be achieved by removing iteratively points from the cloud. The simplest global solution is to randomly select the desired amount of points from the input point cloud. However, doing so, the resulting sampling density is not predictable, and the point cloud features are not preserved. To tackle this issue, a metric is usually proposed to judge the impact of each point's removal on the quality of the surface representation by the simplified point cloud. Points that impact less the surface representation are the first to be removed, and the process continues until either the simplified point cloud contains the desired amount of points, or points can not be removed without producing high errors. The metric can be based on local feature of the point cloud such as the distance from a point to its neighbours and the local normal deviation [119], quadric error metric [156, 158] or the distance to a local MLS approximation of a point set [4]. Successful adaptation of these approaches taking into account the local feature size have been proposed by [91]. Similarly, curvature and torsion of the points have been employed to define a discrete shape operator used for evaluating the points removal impact in [110]. In [188], feature points

are preserved since they are first calculated and the simplification operates only on the non feature points.

Iterative simplification allow the user to define the desired number of points in the output point cloud. They can be very efficient according to the computation of the impact metric which are usually based on differential geometry properties or on the distance from points to a local surface approximation [157].

4.3.2. Clustering

Another major approach for point cloud simplification is to group the points of an input cloud into clusters and only keep an appropriate representative of the cluster. In this case, the two elements of importance are the choice of the clustering method, and the definition of the representative point for each cluster.

Computationally efficient cluster-based global simplification typically use 2D planar or cylindrical grids, and 3D grids to create clusters. All the points lying in the same grid cell are clustered and their centroid is chose to be the representative of the cluster. We call these simple methods downsampling and this approach is widely used in several applications [104, 129, 192, 211] since it is fast, and it produces a simplified point cloud with a regular sampling density. However, it does not preserve the features of the point clouds and in some cases it is preferable that the sampling density of the simplified point cloud reflects the density of the original point cloud. An octree subdivision guided by normals variation [109] or by local feature size [91] can be used to preserve features but many other methods have been proposed to tackle these issues. They highly rely on the efficiency of the clustering method which can be incremental (such as region growing), hierarchical (clusters are recursively split), or based on k-means and its variant [156]. Clusters can be created incrementally with regard to a radius neighbourhood of select points [185]. [136] used mean shift to create hierarchical clusters in a feature space. [212] used clustering techniques in regular grids to preserve key-points and extended their work with the use of hierarchical k-means clustering [213]. Edges and borders of point clouds can be preserved with an proper subdivision of clusters and an appropriate choice of representative [185, 217].

While some methods allow to specify the number of points in the simplified point cloud (such as hierarchical split and merge methods), the majority does not. As a result, clustering solutions generally focus on the error between the simplified point cloud and the original one and offers a more accurate but less efficient point cloud simplification.

4.3.3. Particles

A third major approach for point cloud simplification is to spread a specified amount of points called particles over the original point cloud [195]. Once particles have been initialised, they evolve iteratively under repulsion forces and projectors towards their final location. At each iteration, (1) every particle exerts a repulsive force inside a radius of influence in order to obtain an homogeneous sampling, and (2) particles are projected back onto an approximation of the surface described by the original point cloud to ensure their location accuracy. This scheme has been adapted to produce an adaptative sampling preserving the features of the cloud [156] by weighting the repulsion radius with respect to the features of the initial points.

This method has proven to produce accurate result when the projection operator is defined using quadric error metrics. It is intuitive and can be tuned to provide different simplification ratio over different part of a data set. additionally, particles can even be used to up-sample a point cloud or smooth the data. However, they are only computationally efficient for high simplification rates [156].

4.3.4. Other simplifications approaches

Finally some works that deviate from the three main approaches are present in the literature. As an example, the incremental construction of the simplified point cloud proposed by [138] which is based on the farthest point sampling notion and its extension in [139]. Also, graph based approaches that do not rely on first or second order geometry of the points have been proposed by [40]. Even though these methods are not predominant, the graph based approach seems very promising since it does not require to compute any geometrical feature to obtain an accurate simplification.

4.3.5. Conclusion

In the context of forestry, the data often contains tens of millions of points and the computation of accurate and significant geometrical feature is difficult. Point cloud simplification algorithms seem advantageous for solving the problem of data size, however, in the applicative case of precise measurements from point clouds, keeping the original geometry of the data set is crucial for accuracy.

Therefore, only when point cloud simplification is mandatory, a fine 3D regular grid downsample that is selected.

4.4. Segmentation

In the last part of our work we focused on segmenting individual trees from a t-LiDAR acquired forest plot. Hence we identified the existing methods for point cloud segmentation and derived their strengths and weaknesses for making our methodological choices. Moreover, point cloud segmentation is a crucial problem in a whole part of shape extraction approaches. Point cloud segmentation is also included in numerous methods closely related to our tubular shape extraction (see chapter VI). Therefore we introduce here some major aspects of point cloud segmentation.

As a pre-processing step, point cloud segmentation aims at grouping data points with homogeneous properties into clusters. This step is crucial to separate the different parts of objects present in the data set. Similar to image segmentation, point cloud segmentation algorithms can be divided into different approaches whether they are edge-based, region-based, or attribute-based.

4.4.1. Edge-based segmentation

The idea behind edge-based segmentation is that segments of interest are delineated by edges. Hence, finding the edges leads to a natural segmentation procedure. In [178], the authors follow this approach to perform a segmentation of a point cloud from a range image. The algorithm considers each scan line (profile digitization of a range image) separately to filter the noisy points, and cut the scan line into a set of polylines separated by edges points, hence generating a binary edge map. A minimum spanning tree of the edge map is then computed and filtered in order to thin the detected edges and remove noisy edges.

While edge-based approach relies on an intuitive formulation of the segmentation problem, it is difficult to apply directly on point clouds. Hence, like the work of [178], the application of these algorithms is often restricted to range images and can not be easily adapted to registered point clouds. Moreover, the authors state that edge-based approach does not guaranties a closed contour of the segments, which can result in an under-segmentation of the data.

4.4.2. Region-based segmentation

Region-based segmentations are the counterpart of the edge-based approach. While edge-based approach focus on finding the separation between segments considering local changes of attributes, region-based algorithms tends to identify the homogeneous regions contoured by the edges. These region can be computed either by seeded or unseeded methods.

Unseeded (or top-down) methods for the identification of homogeneous are related to split-and-merge framework. These segmentations start by considering the entire data set as a whole region and computing its homogeneity (on point clouds, the homogeneity is often based on geometric measurements). If the region is not enough homogeneous, it is then split into two or more subregions. This procedure is iteratively of recursively repeated until all extracted regions are homogeneous. The main difficulty of these methods however is to define an homogeneous criteria and to decide how to subdivide a given region. These choices are often delicate and the resulting algorithms often produce under- or over-segmentations.

Seeded (or bottom-up) methods are more widespread in the field of point cloud segmentation. These algorithms grow a region starting from point identified as a seed by incrementally inserting data points that do not break the region homogeneity. In the literature, region homogeneity can be derived from zero, first, or second order geometry (i.e. points position, points normals, and points curvature). A zero order geometry homogeneity is based on the Euclidean distance of the points inside a region: point close one to another are aggregated in an Euclidean clustering [152]. The same idea was used by [124] but was performed thanks to a structuring element. [32] used a variant of the k-means algorithm to perform the corresponding Euclidean segmentation. Integrating the normal vectors to the algorithms, first order region-growing segmentations can be achieved with more precision (e.g. [198]). In this case, the homogeneity criteria is based both on the points distance to each other and on their normal vector consistency. The second order segmentation is achieved by additionally considering the points curvature [176] and principal curvature directions [198] during the region growing.

4.4.3. Attribute segmentation

Since point cloud segmentation aims at clustering points that share common geometrical attributes, this procedure can be viewed as a the solution of a clas-

sical clustering problem. The clustering can hence be performed in the attribute space instead of the original 3D space. Such method has been developed by [62] where the author consider a set of 7 attributes: 3 attributes for the points position, 3 attributes form their normal vector and 1 attribute for the height of the point above its neighbours. A mode seeking algorithm achieves the final clustering and performs the point cloud segmentation.

4.4.4. Specialized segmentation

When the point cloud segmentation is set to be the solution of a particular problem, more adapted approaches are preferred, resulting in specialized solutions. For example, a solution to the problem of differentiating leaves from tree stems inside a t-LiDAR point cloud has been presented in [12] where the segmentation is performed as a classification step: the experience showed that the intensity of the LiDAR signal return is lower on the stem than on the leaves. The authors benefits from this observation by segmenting the histogram of the intensity returns. Another example comes from the need to segment tree crowns from ALS point clouds. Based on the assumption that the upper part of trees resemble to a convex shape, [39] used a marked watershed algorithm to map each tree crown to a catchment.

4.4.5. Conclusion

Segmentation is a major preprocessing step in point cloud reconstruction and analyses. However, segmentation itself does not reconstruct or identify objects. It allows the ulterior algorithms to process only identified segments sharing homogeneous properties and ameliorate their results. As an example, segmentation has claimed to enhance the accuracy of classification algorithms [126], and is a very helpful step for pattern recognition. Besides the advances in this field, point cloud segmentation still encounters limitations due to the presence of occlusion, noise and non-homogeneous sampling in the data set. Depending on their purposes, the segmentation step may also induce some erroneous topological information. As a result, these procedures can not be employed as high level analysis methods for studying the objects present in a data set, but are rather seen as a convenient hint about what parts of the objects have to be focused.

4.5. Conclusion

During t-LiDAR-aided forest inventories, forest plots are often acquired from different positions. Point cloud registration is therefore a mandatory pre-processing step in this application. Since we will be testing our methods in this context, our developments will be confronted to registration artefacts. The other pre-processing steps however appear to be time-consuming and affect the point cloud geometry. Therefore, in our researches we chose to avoid them and focused on the robustness of our algorithms regarding the t-LiDAR point cloud imperfections. This way we aimed at saving resources as well as computing time. However, pre-processing are still attractive, especially point cloud simplification which is recommended confronted to large point clouds. Yet, they will not impact the accuracy of our robust developments.

5. Point cloud modelling

Once point clouds have been preprocessed, a large set of algorithms can be employed to retrieve high level information about the content of the data. One of the main goal in point cloud processing is to provide an approximative model of the objects inside a point cloud. For example, in our studies we aim at extracting tubular models representing trees or industrial elements like pipes. Different modelling approaches have been previously developed and our methodological choices are derived from the analysis of their strengths and weaknesses. Hence, in the following we present some common approaches for point cloud modelling and others that are closely related to our works. We distinguished three main approaches for point cloud modelling depending on the target information to extract: skeletons provide insights about the shape of an object, surfaces gives a complete geometric description of a point cloud, and pattern recognition uses higher level shapes to describe the content of a data set.

5.1. Skeletonization

As stated in our introduction, an object is mainly defined by its shape. Hence, shape analysis is an important domain for the study of a given object. Skeletonization approaches provide a direct estimate of the shape of an object in a simpler and intuitive formulation. This procedure aims at extracting the skeleton of an object which is seen as an equivalent reflection of its shape. Broadly speaking,

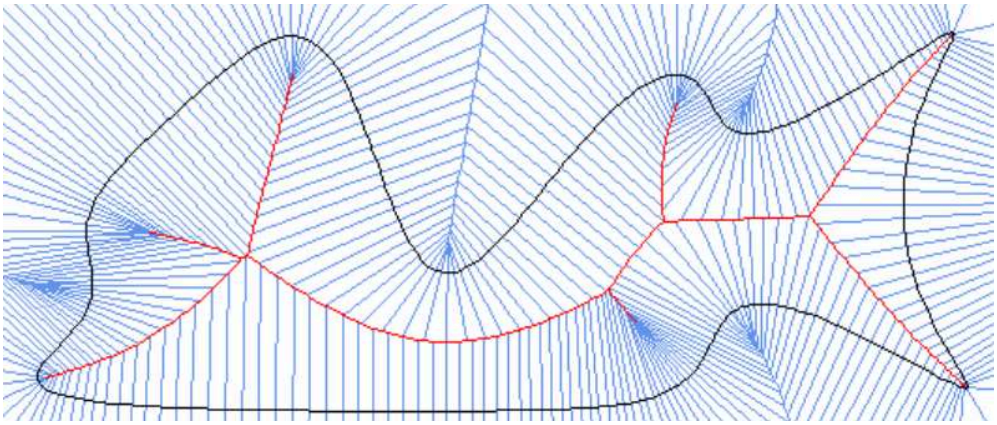


Figure II.24. – The skeleton (red) of an object (black) is computed as the set of interior vertices and edges of the Voronoï diagram (blue).

skeletons are curves that best represent the shape of an object. Hence, the skeletonization procedure reduces a potentially complex data set to a much simpler model. As simple models, skeletons are powerful for analysing the topology of an object, data compression, and pattern recognition.

Multiple definitions of a skeleton have been used through the literature, each of them being equivalent or close to another. It can be defined as the medial axis of an object introduced by [25], or as the shape that is equidistant to the borders of the object. Other definitions include the centres of the maximum balls inside a shape and the centres of the bitangent circles. The skeleton of a point cloud can be derived from various methods depending on the chosen definition. Voronoï-based methods are derived from the centres of maximal balls. Morphological thinning (or *peeling*) methods are induced by the points equidistant to the borders. The rotational symmetry axis (ROSA) and the cross-section analyses with the Hough transform follow the the medial axis definition. Finally, with loose definitions, it is possible to extract an approximated skeleton model on a point using graph representation.

5.1.1. Voronoï-based skeletonization

The Voronoï balls of an input point cloud approximates the definition of maximum balls. As a result, Voronoï diagram can used to extract the skeleton of objects from an input data set: the skeleton is given by the vertices, edges and facets of the Voronoï diagram that are embedded in the object (Fig. II.24). For example, [179] employed this method as a first step for extracting smooth skeleton of trees from range images.

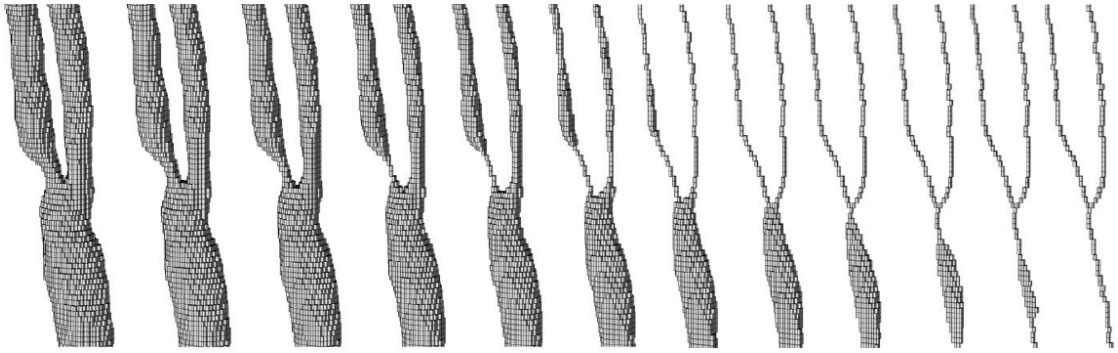


Figure II.25. – Successive iteration of erosion while preserving the homology.

5.1.2. Thinning-based skeletonization

Mathematical morphology offers efficient tools for computing the skeleton of an object embedded in a discrete space. In particular, successive applications of homology preserving erosion of an object achieve the skeleton extraction (Fig. II.25). Since the number of erosion iterations needed to reach a point p is equivalent to the distance transform of p, this iterative *peeling* is similar to the computation of the points equidistant to the object borders. In [68] and [67], the authors first discretized a point cloud with a regular grid and filled the holes present in the voxel domain before retrieving the skeleton with the thinning-based method.

5.1.3. Medial axis-based skeletonization

The medial axis notion in 3D can be related to the centres of a rotational shape. Following this idea, two proposal have been introduced to compute the skeleton of an object with rotational parts, namely the ROSA method and the Hough transform.

The ROSA (ROtational Symmetry Axis) method developed in [191] aims at finding the ROSA points defined as oriented points $p = (x, n)$, with x the position of the point and n its normal, which are "mostly rotationally symmetric about the object". This definition implies that a ROSA point minimises (1) the variance of the angles between n and the normals of P and (2) the square distance to the line extension of the normals of P. The normal n of a ROSA point is first computed by minimising:

$$E_n = \text{var}(< n, n_p >) \tag{II.23}$$

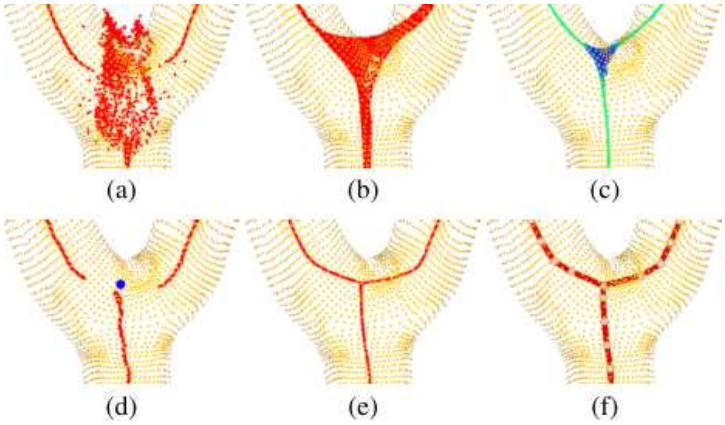


Figure II.26. – The set of ROSA points of an object are shown in red (a) after smoothing operation (b), thinning (c), re-centring and joint collapsing (e,d) to obtain the final 1D curve (f). Adapted from [191].

where var(v) is the variance of a variable v. The position x of the ROSA point is then derived as the minimum of:

$$E_x = \sum_{i=1}^{n} \|(x - p_i) \times n_i\|^2 \tag{II.24}$$

The ROSA points however poorly describe joint regions. As a consequence, several operations, including smoothing and thinning, are applied before the obtention of a smooth 1D skeleton (Fig. II.26).

Aside from the ROSA method, accumulation approach has been employed by Kerautrer et al. in [96] to extract the skeleton of a tubular mesh model which overcomes issues encountered by classical solutions (Fig. II.27). Given a point cloud and the corresponding normal vectors, the author use an accumulator space to observe the convergence of the normals. Then they track the peaks of vote inside the accumulator in successive oriented planes.

5.1.4. Approximated skeletonization

Besides usual definition of a skeleton, curves representative of an object can be seen as accurate approximations. These curves can be extracted from graphs constructed either in a volumetric domain or directly on point clouds.

From on an octree representation of an input point cloud, Bucksch et al. in

Figure II.27. – The skeleton derived from the accumulation method (right) is more robust to occlusion, noise and sampling density obtained by other methods. Adapted from [96].

[34, 35] derive an octree graph based on a robust criterion whether to connect octree nodes by an edge. The authors generate the skeleton as the tree from the octree graph by removing the cycles thanks to determined rules.

Another method employed in [44, 121, 208] directly consider the point cloud for skeleton approximation. In a first step, the Riemannian graph of the point cloud is computed: the vertices of the Riemannian graph are the data points, and the edges are obtained by connecting a point to its neighbours with a weight equal to the euclidean distance that separates them. Then, from an automatically or manually selected root point, the Dijkstra algorithm is used to compute the minimum spanning tree of the graph. In [121] and [44], the vertices of the minimum spanning tree are clustered into bins according to their geodesic distance to the selected root point. The vertices of the skeleton curve are then computed as the centroid of each bin while preserving the bins connections.

5.1.5. Conclusion

Skeletons are widely used in computer vision and image processing since they provide a minimal version of an object. This simplified model allows to identify the topology of an object and can further simplify pattern recognition techniques or classifications. However, it is not an appropriate model when it comes to measure an object. Indeed, aside from the thinning-based approach which requires a point cloud to be expressed in a voxel domain, point cloud skeletonizations do not provide any direct access to measurements. As an example, models obtained from the ROSA and the Hough transform approaches require post processing to retrieve the radius of the tubular object. Also, handling non rotational objects skeleton may be tedious since they are composed of surface patches in addition to curves. Finally, two other issues arise that are the impact of occlusion, noise and outliers on the resulting skeletons. Several post processing have then to be

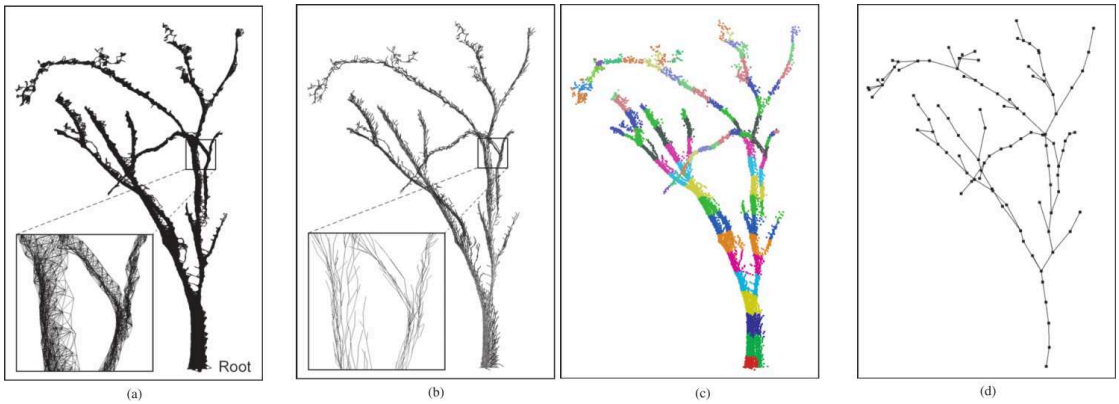


Figure II.28. – The Riemannian graph is computed (a) together with its spanning tree(b) before clustering the vertices into bins (c) to obtain the final skeleton (d). Adapted from [208].

planned for deciding how to reconnect skeleton parts separated by occlusion, and how to prune the skeleton to remove branches induced by noise. Hence, even though skeletonization appears to be very attractive for the reconstruction of tubular shape, these methods still have to be enhanced before providing a satisfying model of an object from which measurements can be extracted.

5.2. Surface reconstruction

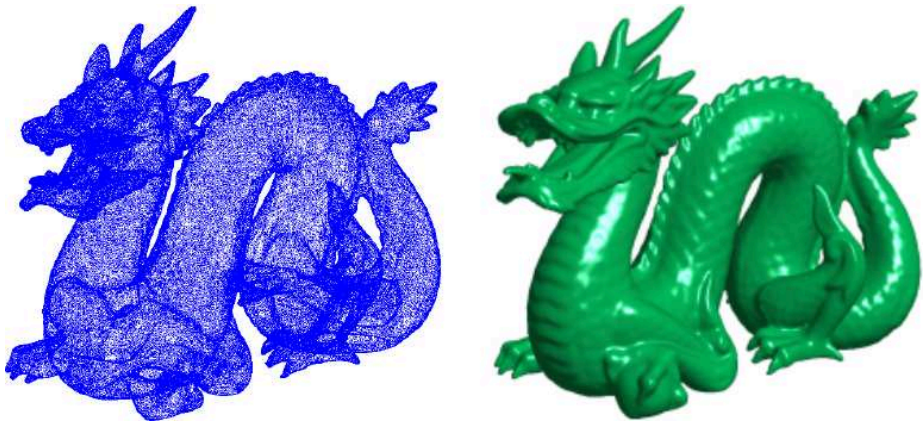


Figure II.29. – Example of surface reconstruction: from a point cloud (left) and its reconstruction as a surface (right). Adapted from [37].

As pointed out in the beginning of this manuscript, a major goal in point cloud processing is to retrieve the surface sampled by the points (Fig. II.29). This is actually one of our major goals. This operation is called surface reconstruction from point cloud. Given a point cloud P sampled on an unknown surface T, surface reconstruction aims at determining a surface S that approximates T based on P. Additionally, information about the sampling process, for example, bounds on the noise magnitude and sampling density can be inferred in the operation [38].

In order to motivate our methodological choices and underline both their links with the state-of-art and novelty, we will now briefly introduce main reconstruction approaches. For more details, we invite the reader to consult the references provided and the review proposed in [17]. Overall, surface reconstruction algorithms can be roughly classified according the surface model used: explicit or implicit models.

5.2.1. Reconstruction using explicit surface models

Explicit models have been widely used for surface reconstruction since they provide direct access to the surface points and proved to be computationally efficient for raw point cloud approximation. Meshes, elevation surface models and B-Splines/NURBS have all been used successfully to achieve surface reconstruction.

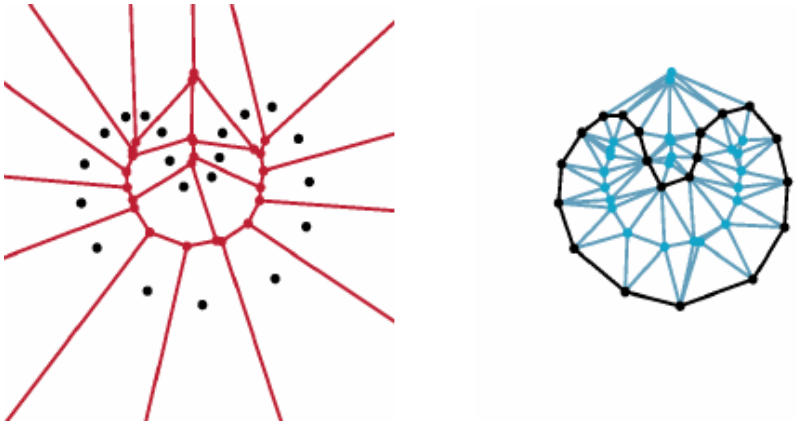


Figure II.30. – Principle of the crust algorithm in 2D. Left: the Voronoï diagram V of the black point cloud P is drawn in red. Right: the Delaunay triangulation of $P \cup V$ is shown in blue while the selected edges are shown in black. Adapted from [7].

Surface reconstruction using mesh models

A straightforward way to reconstruct an explicit surface from a point cloud is to triangulate the points of the cloud and produce a mesh model [117] (Fig. II.34). We now succinctly introduce major algorithms for point cloud meshing. A main approach is based on restrictions of the Delaunay triangulation and its dual the Voronoï diagram [41]. It includes, among others, the crust algorithm [7] and the α-shapes [3, 56] algorithm. As a subset of α-shapes, the ball-pivoting algorithm also uses empty ball properties [19].

As a restricted Delaunay triangulation, the crust algorithm first computes the Voronoï diagram of the point clouds. The next step is to compute the Delaunay triangulation of the point cloud augmented with the Voronoï vertices. The final triangulation is restricted to the edges from the Delaunay triangulation whose circumcircle does not contain any point of the Voronoi diagram (Fig. II.30).

The α-shapes algorithm extracts the subset of simplicies from the Delaunay tetrahedralization that are α-exposed: as illustrated in Figure II.31a and b, a simplex is said to be α exposed if there exist an empty ball of radius α on the simplex. Note that when $\alpha = \inf$, the α shape is the convex hull of the point cloud) (Figure II.31e).

The ball-pivoting algorithm reconstructs a subset of the α-shape model out of a point cloud based on a simple intuitive idea: three points a, b, c of the cloud form a triangle if a ball of a user-specified radius touches them without containing any other point [19]. From a seed triangle verifying the empty ball property, the

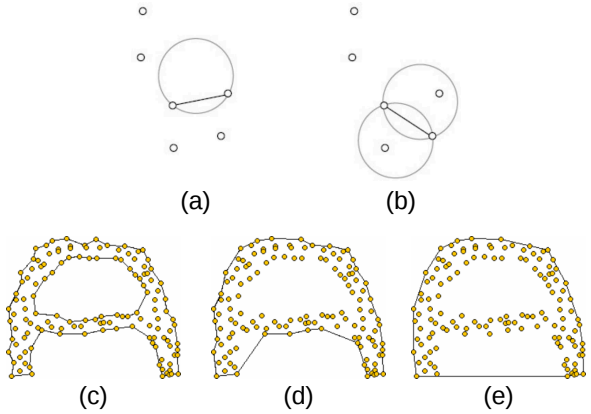


Figure II.31. – Alpha shape definition and results in 2D: (a) and (b) definition of α-exposed simplicies, (c) to (e) show the α-shape of a point cloud for an increasing value of α.

ball is pivoted around an edge (e.g. (a, b)). If the ball encounters another point d during this movement, a new triangle a, b, d is added to the mesh model (Figure II.31c). The algorithm iteratively constructs triangles until all edges have been tried.

Once a point cloud has been modelled with a mesh, it is possible to define a new model based on subdivision surfaces with feature preserving subdivision schemes [82].

Even though surface reconstruction using mesh model appears to be attractive, these methods are highly sensitive to noise and cannot handle occlusion (especially large occlusions). Moreover, the reconstruction of a large point cloud (tens of millions of points) as a mesh model would be computationally expensive and the mesh model would require a high amount of memory resources. Hence these algorithms are not well suited for reconstructing point clouds from t-LiDAR acquisition in natural forest environment.

Parametric models

Surface reconstruction can benefit from the advantages of parametric models. On the one hand elevation surface models have been widely used to locally approximate the point cloud in order to access first and second order geometry. It has also been employed to reconstruct surfaces from point cloud given local parametrizations, but obviously needs ”merging“ functions to provide a full model. On the other hand, B-Splines and NURBS provide global surface reconstruction algorithms. We will now describe them briefly to motivate our choices.

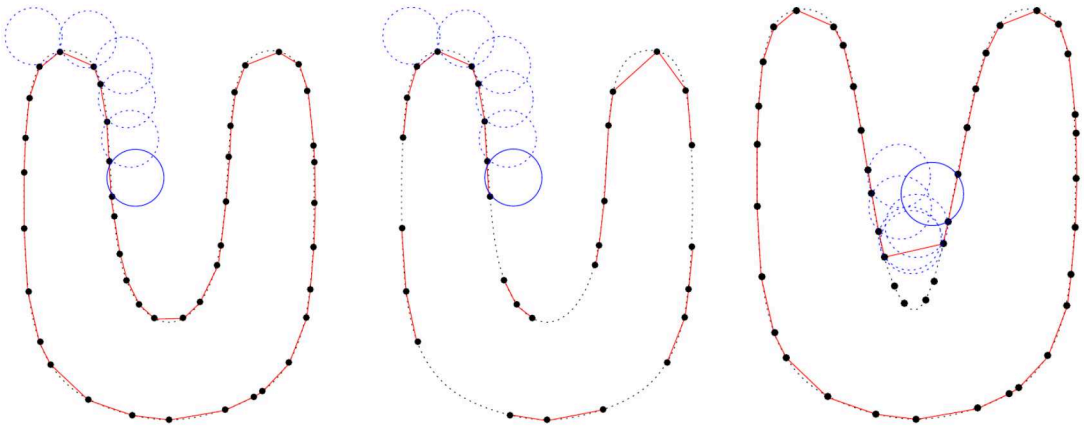


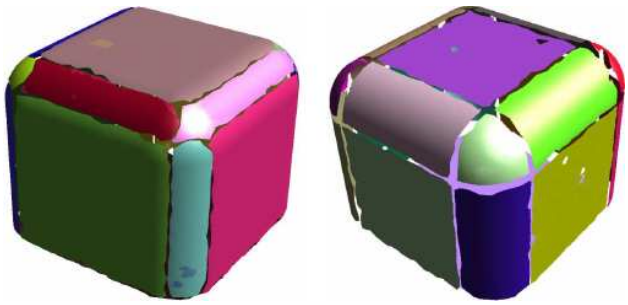
Figure II.32. – Principle of the ball pivoting algorithm in 2D: the ball (blue) pivots along the vertices of the point cloud (black) to generate the reconstruction (red). Adapted from [19].

As a starting point of their algorithm, Ohtake et al. in [149] locally approximate the point cloud with parametric surfaces. Actually, they first determine local frames before using least square optimization to fit a quadratic elevation surface to the points expressed in this local coordinate system. The author further use these local surfaces to estimate an implicit function using compactly supported radial basis functions and merge local approximations.

In [198] the authors use a combination of segmentation and quadric surface fitting to reconstruct a point cloud as a series of quadric surfaces. A first-order segmentation (based on normal vectors) is initially computed and used to refine the estimation of normal vectors and principal curvatures (Figure II.33a). Then, a second-order segmentation (based on principal curvatures) generates segments that are approximated with a quadric surface (Figure II.33b).

NURBS surfaces have also been used to reconstruct surfaces from point clouds [108, 145]. In these mentioned studies, the authors reconstruct a 2.5D point cloud within a single NURB patch. In general, NURBS fitting requires to find an adapted parametrization of the points. A global parametrization can be hard to achieve. However, considering local NURBS patches, tangent planes can be used to easily achieve a local parametrization of each patch. In this case, additional constraints have to be defined to ensure a smooth connection between the different patches.

Surface reconstruction with parametric models alleviates the issue of computer resources encountered by mesh models on large dense point clouds. However, these approaches entail fitting a parametric surface to a point cloud. Usually the approximation is computed using least square optimisation which is sensitive



(a) (b)

Figure II.33. – Results of the first-order segmentation (a) and second-order segmentation (b).

to outliers. Fitting quadratic elevation surfaces or quadric surfaces involves low computational cost, but these struggle to capture the precise geometry when facing sharp features or small details and fail when it comes to restore large holes. NURBS surfaces are more able to reconstruct precisely these details. However NURBS fitting can be tedious when the weight of the control points and the nodes are unknown since the fitting requires to solve a non linear system. Hence, surface reconstruction with parametric models appeared to be inappropriate in the context of surface reconstruction from noisy and occluded t-LiDAR data sets.

5.2.2. Reconstruction using implicit models

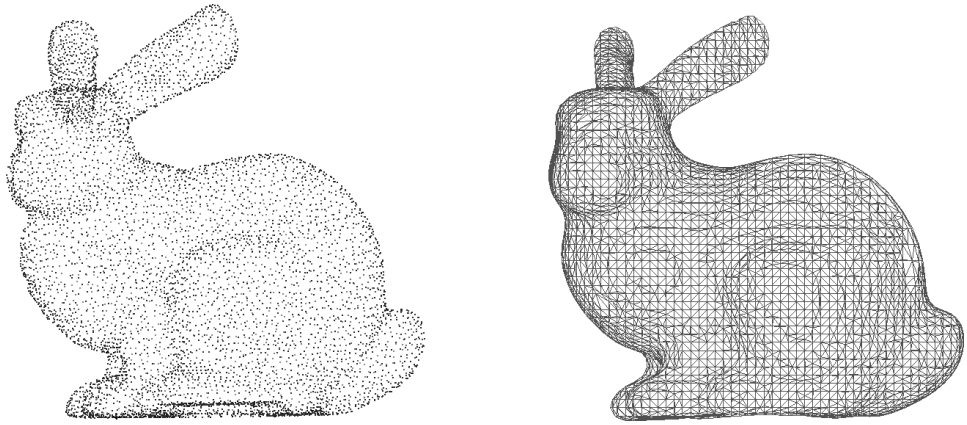


Figure II.34. – Reconstruction of a point cloud (left) as a mesh (right) obtained from the the Poisson reconstruction algorithm [93].

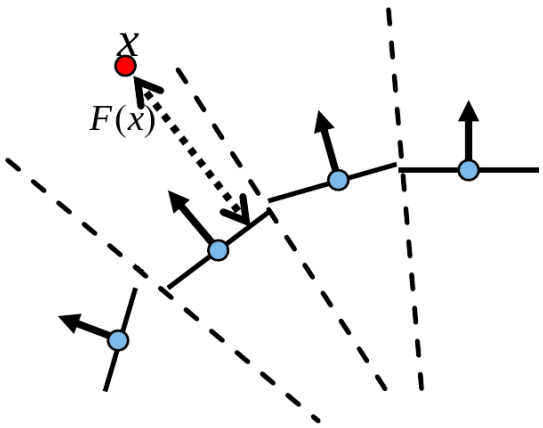


Figure II.35. – SDF approximation: at each data point (blue) the surface is approximated by a tangent plane. The SDF is approximated by the distance from a point (red) to the plane of its closes data point.

The approximation of a point cloud using an implicit model requires to find an implicit function f which zero level set approximates the unknown surface to be reconstructed. The implicit function f is then converted to a surface by extracting its zero level set with the marching cube algorithm or its variations. Several proposal for the definition of the function f can be found. The earlier one is the Signed Distance Function (SDF) defined on a point cloud, but such method only produces piecewise linear surfaces. To alleviate this issue, several authors used the smoothness properties of Radial Basis Functions (RBF). Finally, the Poisson reconstruction focuses on finding an implicit function which gradient approximates points normals.

In an early work, Hoppe et al. in [81] aimed at finding a signed function positive outside the point cloud, negative inside and null on the surface. One function verifying these properties is the SDF of the volume inside the surface. However, since only samples of the surface are known, the SDF itself remains unknown. The authors hence approximates the SDF with the SDF of a local first-order approximation of the surface. Considering that the surface can be locally approximated by planes thanks to normal vectors, the approximate SDF is defined as the distance from a point to its closest approximation plane (Figure II.35). Since the surface is locally approximated with planes, the zero level set of the approximated SDF is only C^1.

The properties of radial basis function have been used to reconstruct smooth surfaces and ensure the continuity of the reconstructed surface. RBF have been largely studied and offer strong interpolation properties [36]. In [37] and [196], the authors approximate data points in a finite dimensional space build using RBF functions centred at a given set of points (called colocations). The implicit

approximating function is then defined as the sum of shifted weighted kernel functions:

$$f(x) = \sum_i \omega_i \phi(\|x - c_i\|) \tag{II.25}$$

where the c_i are the colocations of the ϕ RBF. The function f is then computed so that its zero levelset approximates the point cloud:

$$f(x) = \sum_i \omega_i \phi(\|x - c_i\|) = 0 \text{ if } x \in S \tag{II.26}$$

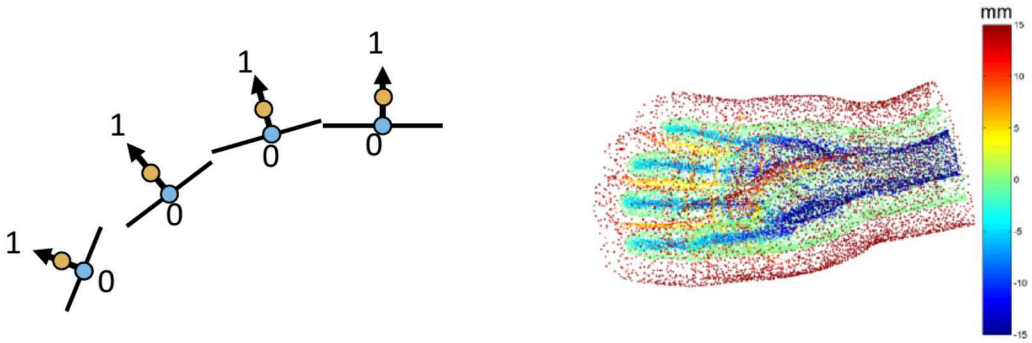
Considering the n points of a point cloud as the colocations of the RBF, this constraint provides a $n \times n$ linear system where the unknowns are the weights ω_i. Unfortunately, such system is degenerated and in order to avoid the trivial solution $\omega = 0$, an additional constraint is required. Following the SDF definition, this additional constraint is given as:

$$f(x + \varepsilon n_i) = \sum_i \omega_i \phi(\|(x + \varepsilon n_i) - c_i\|) = \varepsilon \tag{II.27}$$

that is, the SDF value at a point c_i shifted in/out the surface S along its normal n_i by a value of ε must be ε (Figure II.36). With these n additional constraints, solving the linear system provides a unique solution for the weights ω_i. Many radial basis functions have been studied in the literature and used for surface reconstruction: polyharmonic spline functions, Gaussian functions or compactly supported radial basis functions (CSRBF) such as Wendland's functions [141, 149]. With compactly supported RBF, the linear system becomes sparse which improves the computational time and memory requirements. However, this requires to define the RBF support which can be a challenging task depending on the point cloud sampling density and the level of occlusion [37].

In their work, Ohtake et al. use RBF to obtain a smooth combination of local parametric approximations of the point cloud [149]. In a first step, the authors locally approximate the point cloud with quadratic elevation surfaces S_i. These functions are then blended using CSRBF to obtain the final implicit function f which zero level set corresponds to the surface to be reconstructed.

The author use Wendland's functions ($\phi(r) = (1 - r)_+^4(4r + 1)$) as blending functions to guarantee the smoothness of the final implicit function. To reduce



(a) (b)

Figure II.36. – Additional constraints for the RBF reconstruction: off-surface points are generated along normal vectors (a), and their approximated SDF values are constrained (b). Adapted from [37].

computational time, this surface reconstruction is integrated in a multi-level interpolation scheme [149]. Instead of considering all the points of the cloud, only a subset of points can be selected to achieve better performances [150], and the support of the RBF can be set adaptively in order to make the algorithm robust.

Another major family of implicit surface reconstructions from point cloud is derived from the Poisson equation [93]. These reconstructions approximates the indicator function χ of the volume inside the surface which is 1 inside the surface and 0 outside (Figure II.37). Such a function is discontinuous, however, when convoluted with a smoothing filter, the gradient of the indicator function is zero everywhere except near the surface, where it is equal to the inward surface normal. Considering the points of a cloud as samples of an unknown surface, the algorithm computes the indicator function χ which gradient approximates a vector field V computed from the points normal smoothed using a convolution filter. The divergence operator is applied to solve this problem which becomes a Poisson equation:

$$\Delta\chi = \nabla V \tag{II.28}$$

In order to solve this equation, functions are approximated on a finite basis (convolution of step functions) and solved in the least square sense.

Surface reconstruction using implicit models generate accurate results on clean point clouds. Like mesh models even if they are able to fill small occlusions in the point cloud, they cannot reconstruct a partial shape or a highly occluded point cloud. Hence, these reconstructions cannot be applied as such to t-LiDAR data in

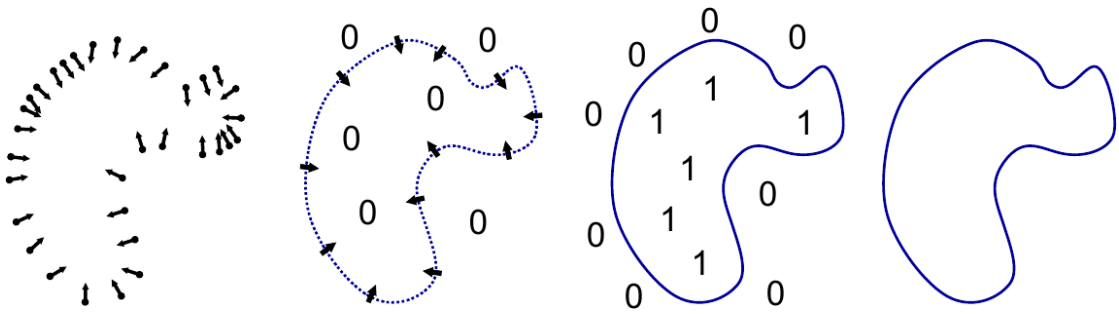


Figure II.37. – Given a point cloud and its normal vectors (left), the Poisson reconstruction aims at finding the 0.5 level-set (right) of indicator function whose gradient is zero except at the surface borders (centre-left) and whose value is zero outside the surface and 1 inside (centre-right). Adapted from [93].

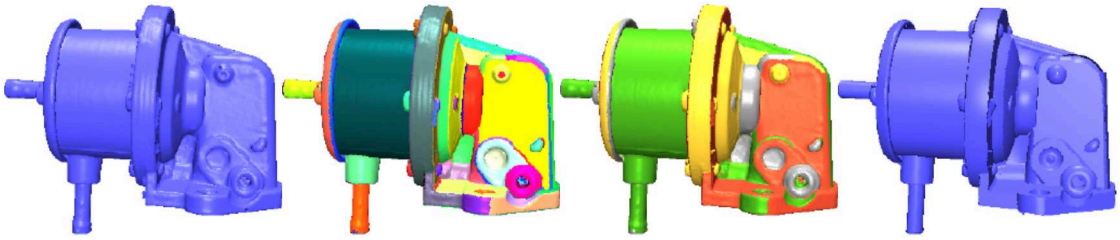
forest environments.

5.2.3. Conclusion

Overall, current surface reconstruction algorithms suffer several limitations when applied on noisy and occluded data. In addition to handling holes, these algorithms are indeed unable to reconstruct a full object if it has only been acquired from a single position (the point cloud is incomplete). Also, surface reconstruction algorithms are designed to reconstruct a single surface (closed or open according to the approach). Objects of interest have thus to be segmented prior their reconstruction. For instance, a set of tubular shapes (such as a set of pipes) need to be identified before being reconstructed. Finally, the shape of the reconstructed surfaces (i.e. tubular shape, cubic shape, general shape, etc) cannot be easily derived from its model. Hence, the reconstructed surfaces do not provide immediate access to shape measurements. These limitations makes the surface reconstruction algorithm unsuitable for our objectives.

5.3. Pattern recognition

When *a priori* knowledge about the shape of a scanned object is available, the surface reconstruction can be turned into a more specific class of problem which is pattern extraction. The objective of pattern extraction is to detect the presence of a given pattern in a dataset. In is a field of *pattern recognition* algorithms whose objective is to identify a common pattern shared by multiple data sets.



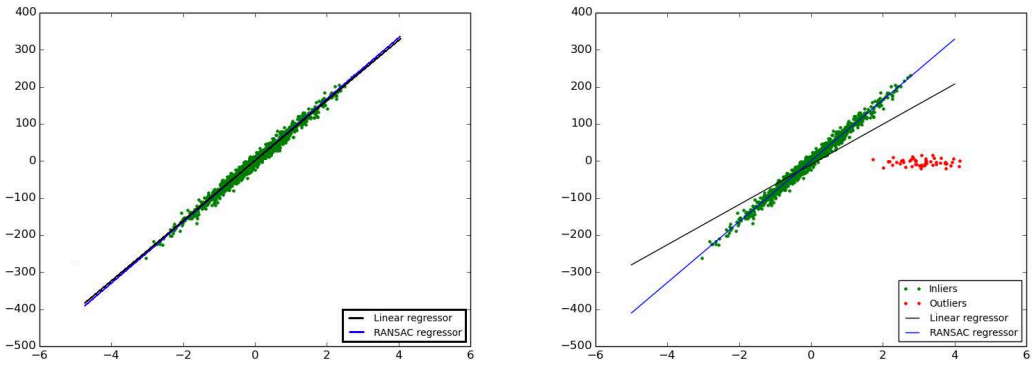
(a) (b) (c) (d)

Figure II.38. – Illustration of primitive extraction on a point cloud: (a) a point cloud of a CAD object, (b) the points are coloured according to the primitive they belong to, (c) the points are coloured according to the type of primitive, and (d) the reconstructed primitives. Adapted from [181].

Pattern extraction is widely used to extract an object defined by a pattern from a data set. It can also achieve a classification of a data set given a set of labelised pattens.

In a general framework, a pattern is an abstraction of an item. It can be viewed as a general model of this item and is often defined from the item characteristics. On a 1D signal, the pattern can be defined from the shape of the signal: On an electrocardiogram, one can look for a specific sequence of signal increasing and decreasing followed by a stable state. This pattern would allow to identify the different phases of a heart beat. A pattern can be defined with regard to the differential geometry of a surface: a pattern defining a crater on an asteroid can be a negative curvature region surrounded by a positive curvature region. In this thesis we address the issue of recognizing and extracting a specific shape in a point cloud. For example, the shapes to be extracted can be planes, spheres, etc. Hence in the later we will consider that the pattern to extract is defined as a general shape and we will refer to this operation as shape extraction.

Shape extraction on point cloud has is of major importance for (1) understanding the content of a data set, and (2) identifying a specific object from a collection of others, and reconstructing it from a point cloud: A wide range of man-made objects designed with CAD or CSG are the result of a combination of primitives (Figure II.38). Hence reverse engineering requires to extract these primitives from a point cloud to retrieve the construction of the scanned object and eventually measure defects. In other contexts, point clouds contain several objects that need to be identified and measured. When a model of an object is available, shape extraction allows to retrieve the objects from the point cloud. For example, the tubular shape model can be used to extract tree stems or pipes from a t-LiDAR point cloud. Moreover, if the model is parametrized, shape extraction



(a) (b)

Figure II.39. – Robustness of least square and RANSAC line fitting. Results without outliers (a) and with outliers (b).

algorithms also provide an accurate estimation of the model's parameters from which precise measurements can be made.

Our research is related to the extraction of tubular shapes and their parameters in point clouds. The size of the point clouds represents a limitation for efficient usages of machine learning approaches (support vector machine, neuronal networks) or graph descriptors. Thus, in the following we will focus on two practical existing approaches that identify a parametric model and its parameters from a point cloud. The first one is based on shape model fitting while the second one uses the Hough transform.

5.3.1. Shape extraction with shape fitting

Given a model of a shape, a straightforward approach for shape extraction can be divided into 3 steps. In a first step, a set of points is selected as candidates. In a second step, shape fitting techniques are applied to estimate the parameters of the model that best approximates the candidate points. Finally, the third step consists in analysing the distance from the candidates to the fitted shape: if the error is less than a defined threshold, the shape is considered to be a valid representation of the point cloud, otherwise, it is discarded. During this third step, the root mean square error (RMSE) of the shape fitting is usually considered. Fitting a given model to a set of points can be done in several ways. However, two methods emerge as classical solutions to this problem, namely least square regression and the random sampling consensus algorithm (RANSAC).

Given a parametric shape model M with k parameters $p_1, p_2, \cdots, p_k$, shape fitting can be posed as an optimisation problem. Indeed, the goal is to find the set of parameters $p'_1, p'_2, \cdots, p'_k$ minimising the distance between the n points $x_1, x_2, \cdots, x_n$ and the model $M(p')$. Expressed in a least square regression fashion, the problem is to find the minimum of the following energy E:

$$E = \sum_{i=0}^{n} \|d(x_i, M(p))\|^2 \tag{II.29}$$

with $d(x_i, M(k)$ is the distance from a point x_i to the model $M(p)$. The energy E reaches its minimum when:

$$\nabla E = 0 \tag{II.30}$$

Thus, solving the previous system of equations for the k parameters provides the desired solution. The least square regression is a very intuitive and widely spread method. However it includes three major limitations. Firstly, this regression is very sensitive to outliers (Fig. II.39). Secondly, when the system of equation (II.30) is not linear (depending on the considered shape and distance), computing the least squares solution involves a higher computational cost and requires a first estimation of the shape close to its final solution . Finally, if the energy E is convex, then the least square regression provides the shape corresponding to the global minimum of E. However, when this condition is not met, the least square method only ensures to provide a solution corresponding to a local minimum of E.

To tackle the limitations of least square regressions, the RANSAC algorithm [63] have been developed, generating multiple variants (e.g. MSAC [194]). RANSAC was developed to be robust to noise and outliers. The main idea of this algorithm is to differentiate inliers and outliers (Fig. II.39): the algorithm fits a shape to a subset of randomly selected data points and identifies the outliers and inliers with respect to the distance from the points to the shape. If the number of inliers associated to the shape is too low, the fitted shape discarded. Otherwise, it is viewed as a good candidate and the distance from inliers to the shape is computed. This procedure is repeated a high number of time starting with a different subset of points, generating a set of candidate shapes. The algorithm's output is the candidate shape that generates the lowest distance to the inliers. Despite its robustness to noise and outliers thanks to their identification, RANSAC does not prevail over least squares regression for several reasons. Indeed, the algorithm requires the user to set different thresholds which influence

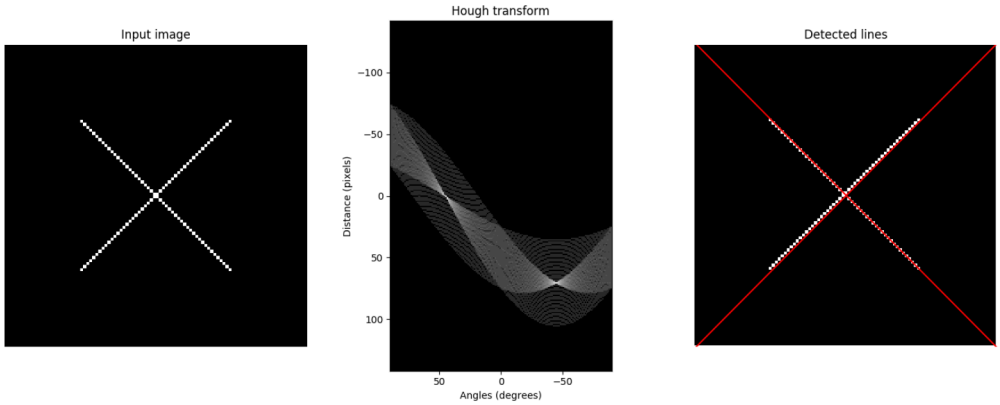


Figure II.40. – Line extraction using Hough transform: (left) input image, (middle) accumulator space, the intensity corresponds to the score of the cells, (right) extracted lines.

the quality of the algorithm. Also, the number of iterations required to obtain an accurate solution can make the algorithm time consuming, even if heuristics have been proposed for an earlier stop of the algorithm. Finally, contrarily to least square regressions, the RANSAC algorithm does not generates an output that is the optimal solution of a formulated problem.

Overall, shape extraction for model fitting requires to identify which points of the data set have to be considered. Such prior segmentation however is still challenging as mentioned in section 4.4. The robustness of the shape fitting algorithms is thus related to the robustness of this segmentation step with regard to occlusion, noise and outliers. Still, least square regression and RANSAC have been used in various applications such as plane extraction for building reconstruction [27], sphere markers detection from t-LiDAR acquired point clouds [215], general point cloud modelling with primitives [113, 181] (Fig. II.38).

5.3.2. Shape extraction with Hough transform

Another approach for shape extraction is the Hough transform (HT) which detects the presence of a given parametric model. The HT is a powerful pattern recognition tool first presented by Hough in [84]. It uses an accumulator to extract a set of occurrences of a shape within a dataset. It was initially devised to recognise straight lines in a picture. Since then, it has been used to detect a wide range of shape models within different types of data [11, 87, 190]. We will now give a general and formal presentation of the HT. However, with the multiple applications and descriptions of the HT, more complete presentations

can be found in several references such as [125, 143].

Roughly speaking, the HT of the shape model is the function which associates the number of data points matching the shape to each potential occurrence of the model. In other words, each data point votes for the set of occurrences that it matches. Let us consider a set of n data points $D = \{d_i,\ i \in 1, \cdots, n\}$ and a mathematical shape model M with m parameters. Let $P = P_1 \times \cdots \times P_m$ be the parameter space with P_j the domain of the j^{th} parameter. Any set of parameters $p \in P$ gives rise to an occurrence $M(p)$ of the model. Let us denote by $f_p(d)$ the binary function testing if a data point d belongs to $M(p)$. Given $k \leq m$, the k-Hough transform is defined as:

$$\begin{array}{rccc} HT_k: & M & \longrightarrow & \mathbb{N}^+ \\ & p & \longmapsto & \sum\limits_{d \in P_k(D)} \mathbb{1}_p(d) \end{array} \tag{II.31}$$

where $P_k(D) \in D^k$ is the set of combinations of k elements among D, and $\mathbb{1}_p(d)$ determines if the tuple $d = (d_1, \cdots, d_k)$ matches the model $M(p)$:

$$\begin{array}{rccc} \mathbb{1}_p(d_1, \cdots, d_k): & \mathcal{D}^k & \longrightarrow & \{0, 1\} \\ & (d_1, \cdots, d_k) & \longmapsto & \prod_{i=0}^{k} f(d_i) \end{array} \tag{II.32}$$

In practice, the parameter space is discretised, and is referred to as Hough space (HS). Each element of the HS represents an occurrence of the model and is associated with a score corresponding to the number of votes (Fig. II.40). Hence the score describes the level of matching between an occurrence of the shape model and the data points. Shape reconstruction is then computed by extracting elements of high score in the HS (Fig. II.40). However, HS analysis is dependent upon the specific shape model and algorithm settings. Therefore, filtering, thresholding and detecting maxima in the HS are key issues and must be adapted to optimise the algorithm for a specific application.

The HT is a powerful approach which reduces a general pattern recognition problem to a simpler discrete space analysis. It does not require an initial guess of the results (e.g. location or number of occurrences) and extracts an exhaustive set of shapes in a single step. Score accumulation tends to be robust to irregular sampling, noise and occlusions, as it is able to identify the most probable occurrence (hence a "full" shape) from partial data. However, it may require a lot of computation time and memory. Like the shape fitting approach, the HT has been successfully applied on a point cloud to detect, among others, planes for building reconstruction [27] (Fig. II.41), circles for tree stem localisation [186], or cylinders for pipe reconstruction [167].

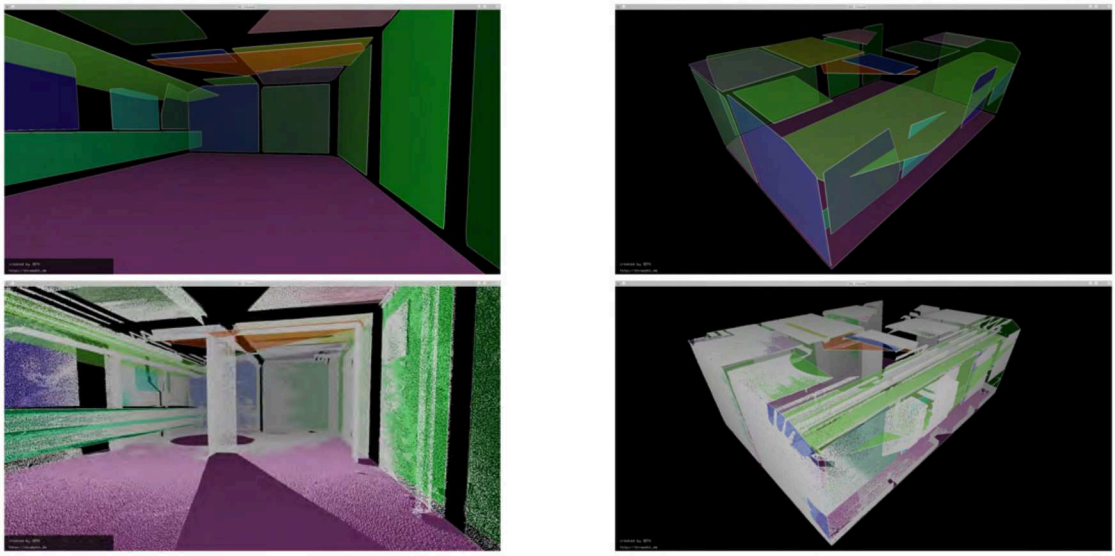


Figure II.41. – Extraction of planes for interior reconstruction. Adapted from [54]

5.3.3. Conclusion

Pattern recognition, and particularly shape extraction uses some small knowledge about the data in order to provide high level information about its content. It is a key element for retro-engineering and automatic measurements. These methods are hence closely related to our problems and we will take advantage of their strengths in our developments.

III. Fast normal vectors estimation

Overview

In chapter II we showed the importance of first order geometry estimation on point clouds. Indeed, numerous algorithms require normal vectors in addition to data points as input. In particular it is a prerequisite for our tubular shape extraction detailed in chapter IV. However, experience showed that existing normal estimation methods are time consuming: most of them necessitate to compute the neighbourhood of a data point before estimating a normal vector. Even though neighbours queries are accelerated with data structures, the repetition of this procedure for each data point makes the process tedious and slow on large point clouds. Therefore, normal estimation constituted a limitation for our developments. We addressed this issue and proposed a normal estimation procedure running in reduced time.

In this chapter we introduce the fast normal vector estimation method we developed. Our approach achieved a time-efficient computation by avoiding pointwise operations. Our work is based on the subdivision of a point cloud into patches of points. The geometry of each patch is then approximated with a local explicit surface model and normal vectors can be efficiently computed for the set of points in the patch. This way we obtained a higher level geometric representation of the point clouds and accelerated further processing. Our development led to the definition of a new data structure called anisotropic octree. In addition to speed up normal estimation, this data structure can be used for classical applications such as neighbours queries. Finally, even though it is not the core of this chapter, we also illustrate the potential of anisotropic octrees for point cloud compression.

1. Introduction

With the current advances in photogrammetry and the availability of instruments recording 3D information (e.g. depth sensors and laser scanning devices from airborne, terrestrial or mobile platforms), point clouds are becoming commonly used in various fields. Data acquired with these tools often contain several tens of millions of points. Processing these large data sets is still a challenge and computing geometric information such as normals or curvature is a real issue. A major difficulty in such processing is the absence of neighbourhood information: unlike meshes, point clouds are unstructured raw data. Consequently, extracting information from a point cloud can be time consuming. Thus, there is a need for new algorithms oriented towards fast point cloud processing.

Normal directions are crucial geometric information for surface analysis. They are a key variable for a large range of algorithms and point clouds analysis approaches [120]. Therefore, normal estimation techniques have been widely studied (see chapter II). However, the estimations of point's normal can be challenging and computationally expensive.

In the present study we propose a new method, following an optimisation-based approach, as a solution to improve the efficiency of normals estimation on large in-core point clouds. Our first specific objective is to avoid point-wise operations. To reach this goal, we developed two distinct procedures that rely on an innovative adaptive subdivision of the point cloud. We propose two novel rules to guide this subdivision in accordance to the local geometric complexity of the point cloud rather than the local density. Our second specific objective is to analyse their efficiency and accuracy in different situations to evaluate their behaviour and the comparative gain in efficiency with currently available algorithms.

2. Methodology

The normal estimation method we developed is built on a simple idea: surfaces are composed of a combination of *uniform* areas (with respect to the geometrical variation and hence, containing no sharp elements) and sharp features. Consequently, the point clouds resulting from remote sensing of these surfaces also contain uniform and sharp parts. The proposed fast normal estimation procedure takes advantage of this characteristic. We avoided costly point-wise fittings by partitioning the input point cloud into patches in an anisotropic division.

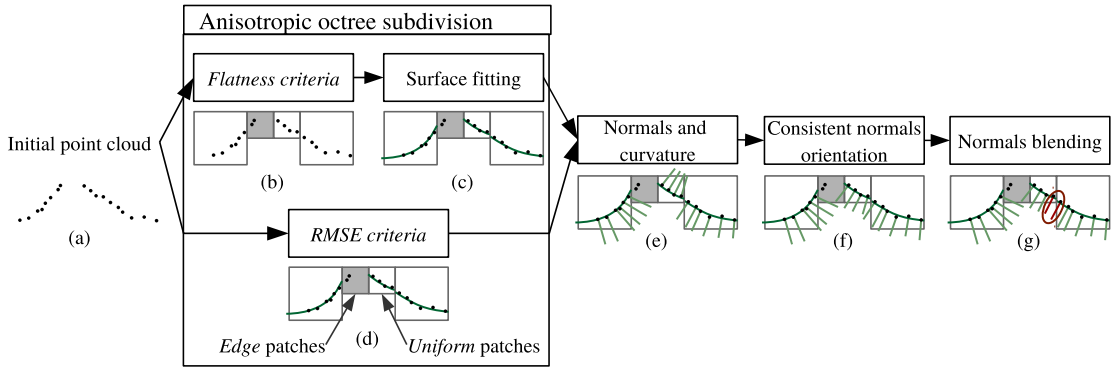
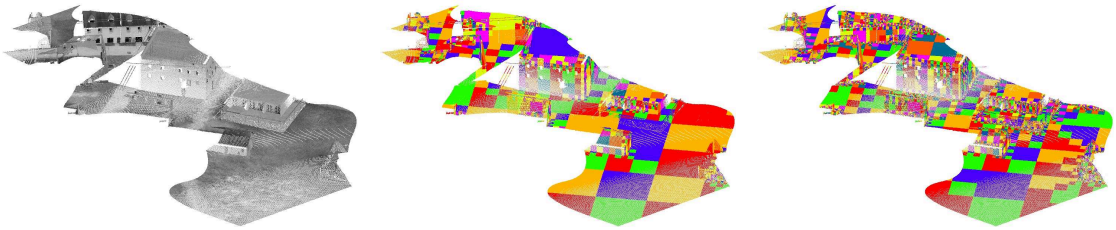


Figure III.1. – Overview of the methodology.

Unlike classical subdivisions that are only guided by the point density, our subdivision takes into account the geometry of the data. Indeed, it is designed to split the point cloud anisotropically according to the set of normals. The fast normal estimation procedure is then applied on patches of points instead of point-wise. More precisely, to achieve our objective efficiently, we developed the methodology summarised by the flow diagram of Figure III.1. This methodology is a compromise between running time and normal directions accuracy. The proposed method improves the running time efficiency based on two main aspects. First, it estimates normal directions at a patch level which lowers the computing time. It should also mitigates the impact of noise on normals estimation on large patches in the sense that with larger areas to approximate comes more information about the underlying surface. Second, it provides a curvature indicator and principal curvature directions, which are important geometric features for further processing. As a side application, the analysis of the point cloud at a patch level can be used for point cloud compression.

The proposed method has four distinct steps. As demonstrated by Zhao et al. (2016), the use of a point cloud subdivision prior to shape fitting shortens the computational time. The first step of our method addresses patches creation by diverting the octree structure from its initial purpose. Unlike Zhao et al. (2916), who use an octree partitioning guided by the point density, our patch partitioning builds an anisotropic subdivision of the data according to its geometry.

We calibrated the octree subdivision of a point cloud on its local uniformity rather than adopting the usual strategy using the point density. Figure 2 illustrates this anisotropic subdivision: areas containing sharp features such as ground-wall, wall-wall or wall-roof junctions are highly subdivided. whereas bare ground or inner walls are less divided and result in larger patches. This *anisotropic octree* recursively divides a point cloud into two sets of patches (Figure III.1b, III.1d): (1) locally smooth patches, and (2) patches containing sharp features (we will



(a) (b) (c)

Figure III.2. – Anisotropic subdivision of an input urban point cloud (left), using σ_3 criterion (centre), or $RMSE$ criterion (right). Sharp features creates small patches while uniform surfaces generates large patches. On an almost constant point density, the octree sudvision still creates patches of various size since it is ruled by geometric complexity.

refer to them as *edge* patches). The recursive subdivision also includes a surface fit for each generated patch (Figure III.1c, III.1d). This way, we generate large patches in even areas and process them at once. Unlike point-wise approaches, our method does not require the computation of a neighbourhood for each point of the input cloud. As a consequence, the complexity of our approach is drastically lower than point-wise approaches. The second step of the method involves estimating normals and curvatures inside each resulting uniform patches using the fitted surfaces (Figure III.1e). Since these surfaces are not oriented, the resulting normal directions are possibly inconsistent across neighbouring patches. Therefore, the third step is designed to improve the coherence of the normal field. To do so, we take advantage of the fast node neighbourhood access given by the octree structure to generate a consistent orientation of normals (Figure III.1f). Eventually, the forth step blends the computed normals and curvatures along nodes boundaries in order to guarantee the smoothness of the resulting normal field (Figure III.1g).

2.1. Anisotropic subdivision

The first step of our method divides the point cloud along an octree subdivision guided by the local uniformity of the point cloud. Starting from the root of the octree $\mathcal{O}$, which contains the entire point cloud, nodes are split into eight octant patches until a uniformity criterion is satisfied.

We developed two procedures to create the anisotropic subdivision (Figure III.2). Each of them uses a different criterion to stop the recursive subdivision. The first procedure stops the subdivision when the flatness of the patches is sufficient (Figure III.1b): the corresponding criterion is based on a flatness index

(Figure III.3). The second procedure stops the subdivision as soon as the patches can be accurately approximated by a surface model (Figure III.1d): the corresponding criterion thus is based on the modelling error.

Both criteria are designed to avoid an over-segmentation of the initial point cloud. Indeed, only patches containing sharp features will be further divided, hence generating large patches for uniform areas, and small patches for sharp features, that is, an anisotropic subdivision. The two procedures have different strengths and balance running time and accuracy, as discussed in the results and the discussion sections. While the first procedure is based on flatness, it requires less computations but captures fewer details on the point clouds. Indeed, Figure III.2 shows that the σ_3 criterion is more permissive regarding the presence of sharp features and creates patches containing a small amount of sharp features such as ground-wall junctions. The second procedure based on modelling error is more accurate, but involves more calculations. While the verification of the first criterion requires the use of plane fitting operations, quadratic surface fitting is necessary for the evaluation of the second criterion.

2.1.1. Local plane fitting

Plane fitting

Let $o \in \mathcal{O}$ be an octree node, $\{p_1, p_2, \cdots, p_m\}$ be the points contained in o, and $\overline{p}$ be their centroid. It has been proven that computing the least square fitted plane P going through $\overline{p}$ and approximating the points p_i can be obtained by a centred PCA of the p_i [156]. Indeed, such a plane is completely determined by its normal $\vec{n}$. Hence, least square fitting is equivalent to minimising the following least square error:

$$\begin{aligned} \sum_{i=1}^{m} \mathrm{d}(p_i, P)^2 &= \sum_{i=0}^{m} \left(\frac{(p_i - \overline{p}) \cdot \vec{n}}{\|\vec{n}\|} \right)^2 \\ &= \frac{1}{\|\vec{n}\|^2} \vec{n}^t \times \mathrm{Cov}(\overline{p_i}) \times \vec{n} \end{aligned} \tag{III.1}$$

where Cov denotes the covariance matrix, and $\overline{p_i} = p_i - \overline{p}$. Given $\lambda_1 \geqslant \lambda_2 \geqslant \lambda_3$ the eigenvalues of the covariance matrix, and $\vec{v_1}, \vec{v_2}, \vec{v_3}$ the associated eigenvectors, the fitted plane P is actually the plane containing $\overline{p}$ and whose normal direction is $\vec{v_3}$.

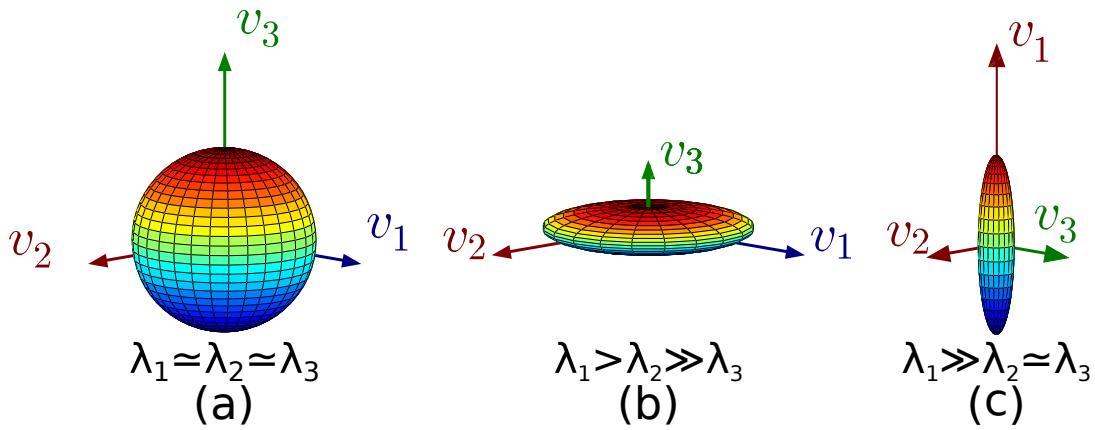


Figure III.3. – Different point cloud dispersion represented as ellipsoids and corresponding PCA results. Eigenvectors are scaled according the eigenvalue. The colours represents the height map of each ellipsoid for a better appreciation of the shape.

Flatness index

The residuals of the PCA plane fitting are given by λ_3. Thus these residuals may be considered as a flatness indicator. However, λ_3 not only depends on the number of points considered, but also on the size of the point cloud. Hence we selected an indicator σ_3 called surface variation [156], providing an homogeneous, orientation, and scale independent indicator:

$$\sigma_3 = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \tag{III.2}$$

From the PCA properties, σ_3 can be seen as the ratio of inertia kept by the plane normal over the total inertia of the point set. This normalisation produces an index $\sigma_3 \in \left[0, \frac{1}{3}\right]$ that is independent of the patch size and of the number of points considered: the lower σ_3, the flatter the corresponding sampled surface (Figure III.3b). This index can be applied as a criterion to regulate the anisotropic recursive subdivision.

2.1.2. Quadratic surface fitting

The second procedure models patches as quadratic surfaces for a closer approximation of their geometry. Computing a best-fit surface requires a local frame. An efficient way to obtain this system is to compute an approximate tangent plane using the previously described PCA plane fitting, and to consider the

orthogonal resulting basis $B = (\vec{v_1}, \vec{v_2}, \vec{v_3})$, along with $\overline{p}$ its origin.

Given a local frame and a point cloud $\mathcal{P} = \{p_1, p_2, \cdots, p_m\}$ expressed in this basis, we use the classical least square method ([163]) to fit an elevation surface S of equation $S(x, y) = a^2 + bxy + cy^2 + dx + ey + f$ minimising the following distance:

$$d(S, \mathcal{P}) = \sum_{i=1}^{n} (z_i - S(x_i, y_i))^2 \tag{III.3}$$

The accuracy of this fitting procedure can be evaluated with the corresponding root mean square error (RMSE), which can be used as a criterion to guide the octree subdivision.

2.1.3. Anisotropic subdivision

Let us now introduce our anisotropic octree subdivision. Whereas standard octree subdivision iteratively divides voxels according to the relative density of points, we control the subdivision and eventually stop it according to either of the two criteria presented: (1) the σ_3 criterion, and (2) the $RMSE$ criterion. The octree creation using the σ_3 criterion performs a PCA plane fitting and stops the subdivision according to a threshold over σ_3 (Figure III.1b). In the $RMSE$ criterion, after the initial PCA plane fitting a quadratic surface is fitted and subdivision is stopped according to a threshold over the resulting $RMSE$ (Figure III.1d).

The σ_3 criterion therefore requires less computations and as resulting patches have a σ_3 value lesser than a given threshold, they are relatively flat and do not contain folds or sharp features. The $RMSE$ criterion requires more calculations since both PCA plane fitting and quadratic surface fitting are computed at each level of the recursive subdivision. However, this criterion assesses more accurately the patches and improves the anisotropy of the resulting subdivision.

Starting from this subdivision, each patch is them modelled by an elevation surface to compute normals. Whereas the $RMSE$ criterion directly provides an approximating surface (Figure III.1d), in the case of the σ_3 criterion, we eventually fit a quadratic surface in the least squares sense over each patch. (Figure III.1c).

Regardless of the criterion applied, we set two additional conditions for a

patch to be split. At critically small scales, point clouds no longer represent the sampled surfaces. Therefore we use a threshold on the patches dimensions D_{min} to avoid dividing further small nodes. Finally, since we use PCA to locally fit a plane, we set a threshold n_{min} on the number of points contained within the patch to ensure a reliable geometric fitting.

As a result of the recursive binary space division, two main reasons exist for an octree node to include a non uniform patch that cannot be modelled by a surface. The first reason is that nodes may contain less than n_{min} points. This happens: (1) mainly when a node containing an edge is recursively split without reaching the desired level of smoothness, or (2) when a node splitting isolates a few points inside an octant, even if the other nodes reach the smoothness criteria. The second reason for a node not to contain a uniform patch is that the patch is more likely to represent a 3D curve than a surface (Figure III.3c). In order to detect these patches, we use another indicator derived from the PCA plane fitting. These *edge* patches are detected by using the following ratio:

$$\sigma_2 = \frac{\lambda_2}{\lambda_1 + \lambda_2 + \lambda_3} \tag{III.4}$$

No surface can be fitted on *edge* patches. Similarly, patches not containing enough points are not modelled by a surface since they may contain a sharp feature that cannot be accurately approximated by a quadratic model (e.g. several piecewise surface). In both cases, these patches are ignored in the later procedure.

2.2. Normals estimation and curvature indicator

Once uniform patches have been obtained from the anisotropic octree and modelled by a surface S, normals can be estimated in an efficient manner. The normal direction at a point p_i of the patch is approximated by the normal of its projection p'_i onto the surface S. Since the surface equation is known, obtaining the normal $\overrightarrow{n}$ is straight forward. However, at this point, the orientation of the fitted surface cannot be assessed (i.e. discriminating interior and exterior space), and neither do the normal directions.

Similarly to normal directions, principal curvatures can be derived from the fitted surface equation. The first and second fundamental forms of each surface are computed for each point p_i as the curvature of their projection on the fitted surface p'_i. Hence, the curvature tensor can be computed to obtain the corresponding eigenvalues and eigenvectors. However, because the fitted surfaces are quadratic, and since the point cloud division generates uniform patches, the cur-

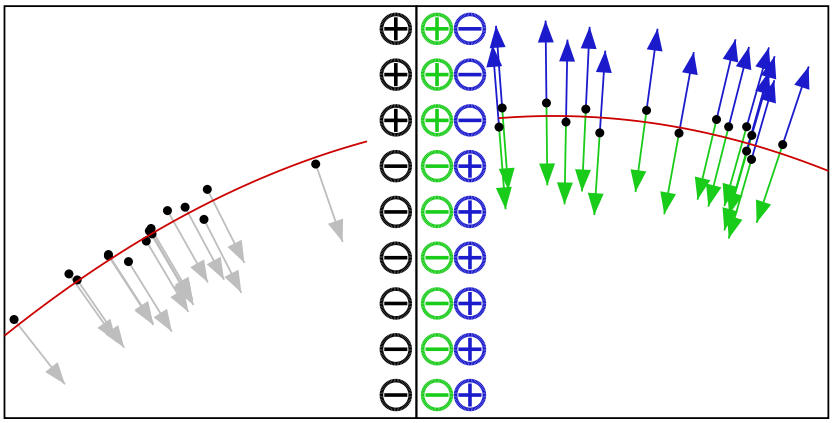


Figure III.4. – Surfaces are oriented by considering the sign of implicit surfaces at points sampled on the face of contact between neighbour nodes.

vature may not be captured with high precision. The resulting eigenvalues of the tensor thus differ from the actual main curvatures of the surface. Nonetheless, this procedure provides a potential indicator of the curvature which can be used to analyse the changes of its value across the point cloud.

2.3. Consistent normal orientation

In order to enhance the consistency of the normal directions, we propose a way to locally orient surfaces. Starting from a patch, neighbouring patches are probed according to their adjacency to propagate a consistent orientation (two patches are considered to be neighbours if their corresponding octree nodes share a common face). Let N_1 and N_2 be two neighbour nodes, S_1 and S_2 be the surface fitted in each node, and f the face shared by N_1 and N_2. In order to orient N_2 relatively to N_1, a set of m points are sampled uniformly on the face f. Now, S_1 and S_2 can be seen as the surfaces corresponding to the zero level set of two implicit functions I_1 and I_2 corresponding to their quadratic equation.. At each point p_i sampled on f, the signs of $I_1(p_i)$ and $I_2(p_i)$ is computed. The orientation of N_2 is set to minimise the difference of signs at the sampled points (Figure III.4).

2.4. Normals blending

Since patches are analysed separately, the junction between the fitted surfaces of two neighbouring patches are discontinuous. Thus, two points inside a small neighbourhood radius may have divergent normals. To minimise this potential

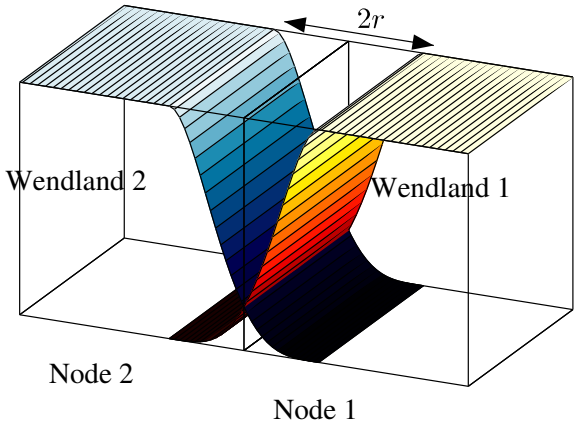


Figure III.5. – Normals directions close to a node faces are weighted with the neighbours node surface. The weights are computed as a Wendland function around the faces of contact.

bias, we propose a way to blend the normal direction at the nodes junctions taking into account the local information of multiple surfaces. Note N_1, N_2 two neighbouring nodes of size d_1 and d_2 and f the face they share. We define a radius of influence r:

$$r = \frac{min(d_1, d_2)}{3} \tag{III.5}$$

around f in which normals will be blended (Figure III.5). At each point p, one or several nodes and their surfaces S_i will then influence the estimation of its normal direction. The normal direction $\overrightarrow{\mathcal{N}}_p$ at p will be ultimately estimated as a weighted sum of the normals $\overrightarrow{n_i}$ at p on the set of surfaces S_i:

$$\overrightarrow{\mathcal{N}}_p = \sum_i \omega_i \overrightarrow{n_i} \tag{III.6}$$

where $\overrightarrow{n_i}$ is the normal at p_i estimated on the surface S_i, and the weights ω_i are computed based a Wendland's function of the distance d_i from p to f_i (Figure III.5):

$$\omega_i(d) = \begin{cases} (1-d)_+^4(4d+1) & \text{if } d < r \\ 0 & \text{else} \end{cases} \tag{III.7}$$

In order to blend local estimates smoothly inside a limited radius, we choose to weight normals along boundaries using Wendland's functions. Indeed, such functions are smooth (to any given degree of derivability) while having a compact support. Linear interpolation is clearly not smooth enough whereas Gaussians functions are not compactly supported (whereas we intend to blend estimates

only along the boundary).

2.5. Point cloud compression

The presented anisotropic octrees provide a quadratic surface patches-based representation of a point cloud. Since patches approximate several data points, fewer patches are necessary to reproduce a point cloud. We used this property and developed a point cloud compression by only considering surface patches.

We propose to achieve the point cloud compression by encoding, in a file, the anisotropic octree structure and the associated surfaces. As a header, the file includes the bounding cube of the point cloud is first represented by 9 floats. Internal nodes are represented by an empty line and empty nodes are represented by a character string. For each leaf node, the file stores:

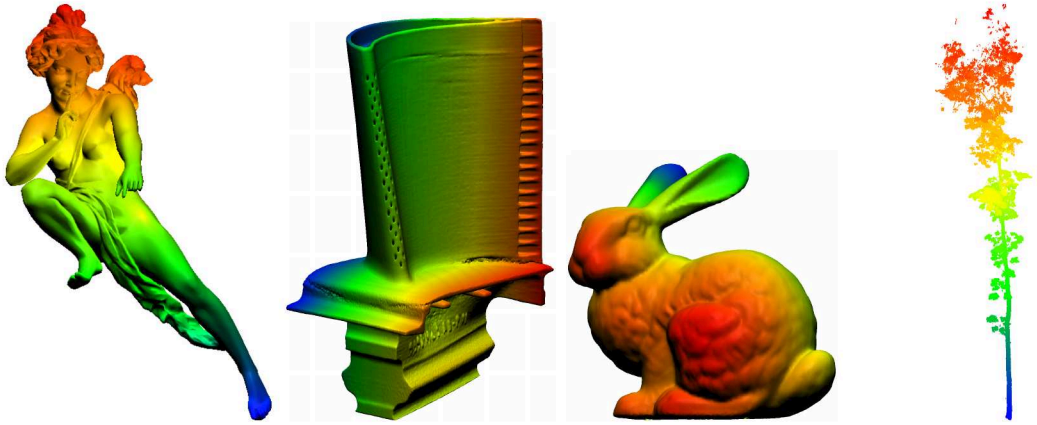
— the origin of the local surface patch (3 floats)
— the vectors of the local basis (9 floats)
— the parameters of the quadratic surface S (6 floats)
— the bounding rectangle B of the points in the local basis (4 floats)
— the number n_{node} of points inside the node (1 integer)

The storage of a leaf node thus requires $23 \times 4 = 92$ bytes. Each data point is originally stored by $3 \times 4 = 12$ bytes. Therefore, the compression becomes advantageous when a surface patch approximates 8 points or more. The decompression is straightforward: surface patches are loaded, and for each patch n_{nodes} points are uniformly sampled on S inside the bounding rectangle B.

The compression rate depends on the criteria selected during anisotropic octree construction (threshold on σ_3 or $RMSE$). However, because patches only approximate the input point cloud, the proposed compression method includes data loss. Therefore, higher thresholds lead to higher compression with higher errors.

3. Results

We tested our methodology on six different point clouds with various characteristics and number of points (Figure III.6). These models are indeed representative of a large range of applications: arts, forestry, urban environments and industrial applications. Additionally, while the *Lucy* model contains a higher number of points, the *Angel* model shows variations in point sampling where



(a) (b) (c) (d)

Figure III.6. – Models used to tests our methods. From left to right: $Angel$, $Blade$, $Bunny$, $Tree$ ($Urban$ and $Lucy$ models are shown in Figure III.2 and III.8). Colours are only used to emphasise the point cloud rendering.

	M	CCP	CCQ	CCT	$RMSE$	σ_3
Median	1.7	2.4	1.7	0.6	3.2	6.5

Table III.1. – Average median error for Meshlab (M), CloudCompare's plane fitting (CCP), quadratic fitting (CCQ) and triangulation (CCT), $RMSE$ octree, and σ_3 octree.

the point cloud is denser around the details of the statue. Note also that the *Tree* point cloud contains a little noise, mainly around the foliage due to the particular limitations of TLS instruments in forest environments. Four out of six models (*Angel*, *Bunny*, *Blade*, and *Lucy*) are accessible from the Stanford 3D Scanning Repository, or the Large Geometric Models Archive of Georgia Institute of Technology. They were available as a mesh with known normal directions which were used as reference directions. The other two point clouds represent a tree (acquired with terrestrial laser scanning), and a simulated urban environment. These six data sets were used to analyse both the running time efficiency and the accuracy of our method. Results from our method were compared with those obtained by two geometric modelling pieces of software providing normals estimation procedures. In Meshlab, the normals estimation is based on a point-wise k-neighbours operation. CloudCompare includes three different methods: point-wise local plane fitting, point-wise local quadratic surface fitting, and triangulation. Our anisotropic octree subdivision with both criteria (σ_3 and $RMSE$), were also evaluated for a total of six different methods.

The running time for all six normal estimation methods on the six point clouds are presented in Figure III.7 (note the logarithmic scale) and the normal estima-

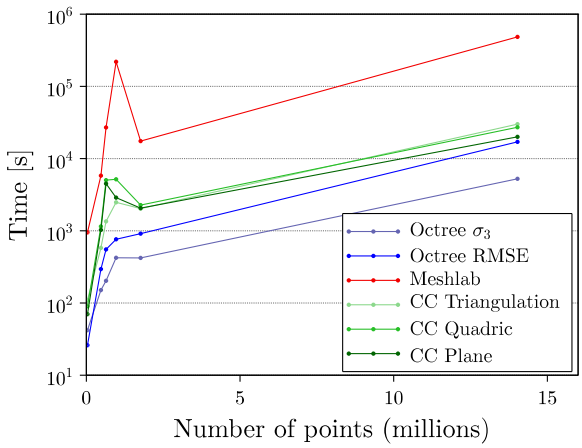


Figure III.7. – Comparison of computational time for normal directions estimation using different methods.

tion error is given in Table III.1. We ran the tests on an Intel© Core™ i7-4790 CPU (3.60 GHz). Our methods outperformed the methods in Meshlab and CloudCompare on all tested point clouds in terms of running time. More specifically, the σ_3 octree was faster than the $RMSE$ octree. These results can be explained by (1) the higher tolerance of the σ_3 criterion to sharp features which induces less subdivisions of the point cloud, and (2) the need to compute the quadratic surface fitting at each level of recursion when using the $RMSE$ criterion, whereas this surface fitting is only computed on the octree leaves when the σ_3 criterion is selected.

The complexity of the anisotropic subdivision is related to the size and the local uniformity of the point cloud. In contrast, since Meshlab and CloudCompare perform point-wise operations, the resulting complexity is then dependent on the number of points in the cloud. In point wise optimisation-based approach, for each point, a neighbour query is required before fitting a surface to the extracted neighbourhood (either with a plane fitting, a quadric surface fitting, or a plane fitting followed by a quadratic surface fitting). Such neighbour finding procedure can be achieved in $\mathcal{O}(log(n))$. Denoting by F the complexity of the chosen fitting procedure and S the complexity of the accelerating structure creation, the overall complexity of point-wise approaches is thus $\mathcal{O}(S + (n * (log(n) + F)))$.

The complexity of our methodology, in the worst case, is that of the octree creation (that is $S = \mathcal{O}(n(d+1)$ where d is the depth of the octree) plus a PCA plane fitting and a surface fitting in each node of the octree. The overall complexity of the normal estimation in an anisotropic octree containing M nodes is thus $\mathcal{O}(S + (M * (PCA + F)))$. In the vast majority of point clouds, $M \ll n$.

(a) (b)

Figure III.8. – Estimated normal directions and curvature on the *lucy* model using the $RMSE$ criterion. Point cloud is rendered using estimated normal directions (left), and curvature indicator (right).

Hence the low complexity of our approach compared to point-wise approaches witnessed by empirical results.

When comparing computing time, we found that the CloudCompare triangulation method is, on average, the most efficient concurrent of our methods. The $RMSE$ anisotropic octree subdivision speeds up, on average, the normals estimation by a factor 2.65 in comparison with the CloudCompare triangulation. This factor rises to 4.92 when considering the σ_3 octree. Overall, the σ_3 octree performs in average 2.09 times faster than the $RMSE$ octree.

We also analysed the normal estimation accuracy and computed the curvature indicators for all six methods on four of the point clouds: the *Angel*, *Bunny*, *Blade*, and *Lucy* models (Figure III.8). To do so, we considered the normals of the mesh models as the reference. Hence, we evaluated the estimation error as the angle between the reference normals and the estimated ones (Figure III.9). Let us also point out that no normal vectors were estimated on the points identified as belonging to *edge* patches (corresponding to sharp features). These points are actually discarded during the comparison procedure. The $RMSE$ octree produces an error distribution similar to the plane fitting and triangulation methods

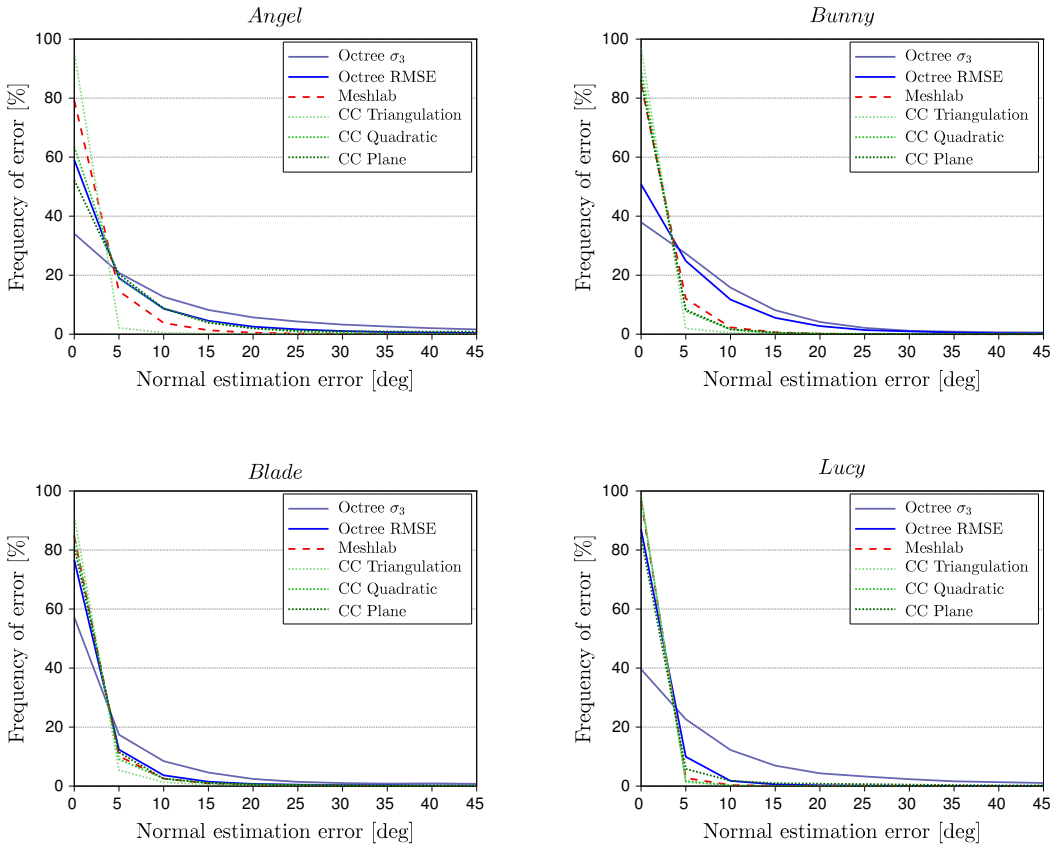


Figure III.9. – Distribution of error angle between original model normals and estimated normals for different methods. Above 45 degrees, the frequency continues to decrease for each method.

from CloudCompare but with higher errors on the *Bunny* model. The σ_3 octree however is less accurate with error distributed on a larger range. Average median errors are summarised in table III.1. As illustrated in figure III.8, even though the values of the curvature indicator differs from the actual Gaussian curvature, the variations of this indicator are actually relevant and capture the reliefs of the statue.

Figure III.10 illustrates the differences between each method, including the tree and urban data sets. A pattern emerges as the anisotropic octree σ_3 induces higher differences for each compared method, and for the majority of data set. Overall, the difference between Meshlab and CloudCompare is regular, and the octree $RMSE$ presents low difference variations with these methods. However, the data sets characteristics influence the resulting differences as shown in the *Tree* and *Angel* data sets. The differences observed on the *Tree* data set are due

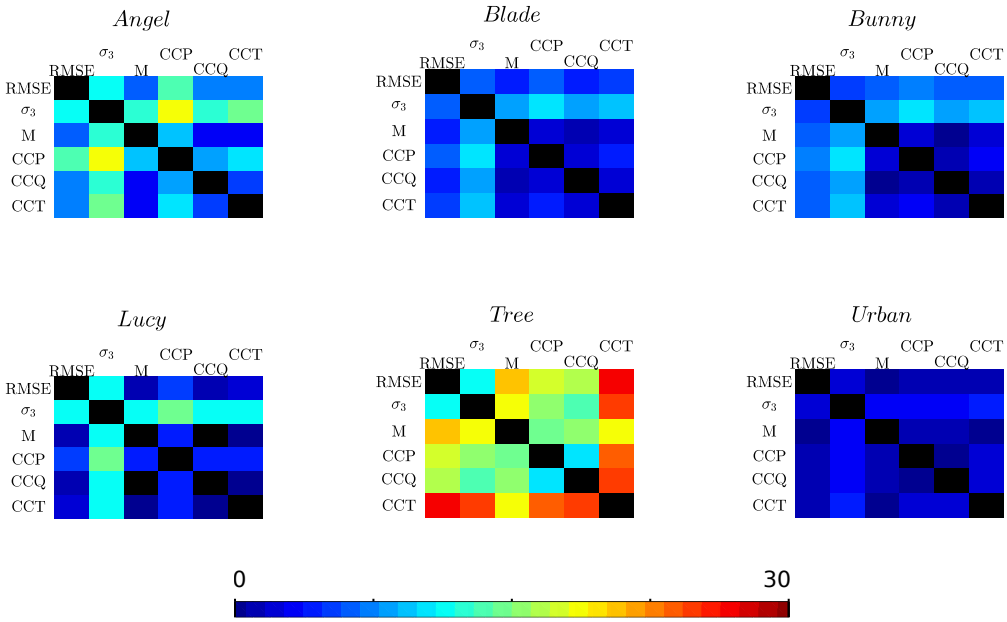


Figure III.10. – Difference of normal directions estimation in degres between the $RMSE$ octree ($RMSE$), the σ_3 octree (σ_3), Meshlab (M) and CloudCompare's plane fitting (CCP), quadratic fitting (CCQ) and triangulation (CCT).

to the high geometric complexity of the analysed object, as well as the occlusion effect inherent to terrestrial laser scanning. The *Angel* point cloud in turn has a non homogeneity of the point sampling on the surface which may impact the neighbourhood search operations. On the contrary, the *Urban* point cloud present lowest inter-methods differences including the σ_3 octree. This is explained by the simplicity of the sampled scene: the point cloud is mainly composed of flat surfaces such as soil, walls, and roofs.

Finally, we evaluated the compression rate offered by the anisotropic octrees (Figure III.11). Table III.2 shows that the compression rate is higher for point clouds containing large flat areas. This result was expected since these areas are compressed as a single quadratic surface patch, whereas more complex surfaces require more patches to be described accurately.

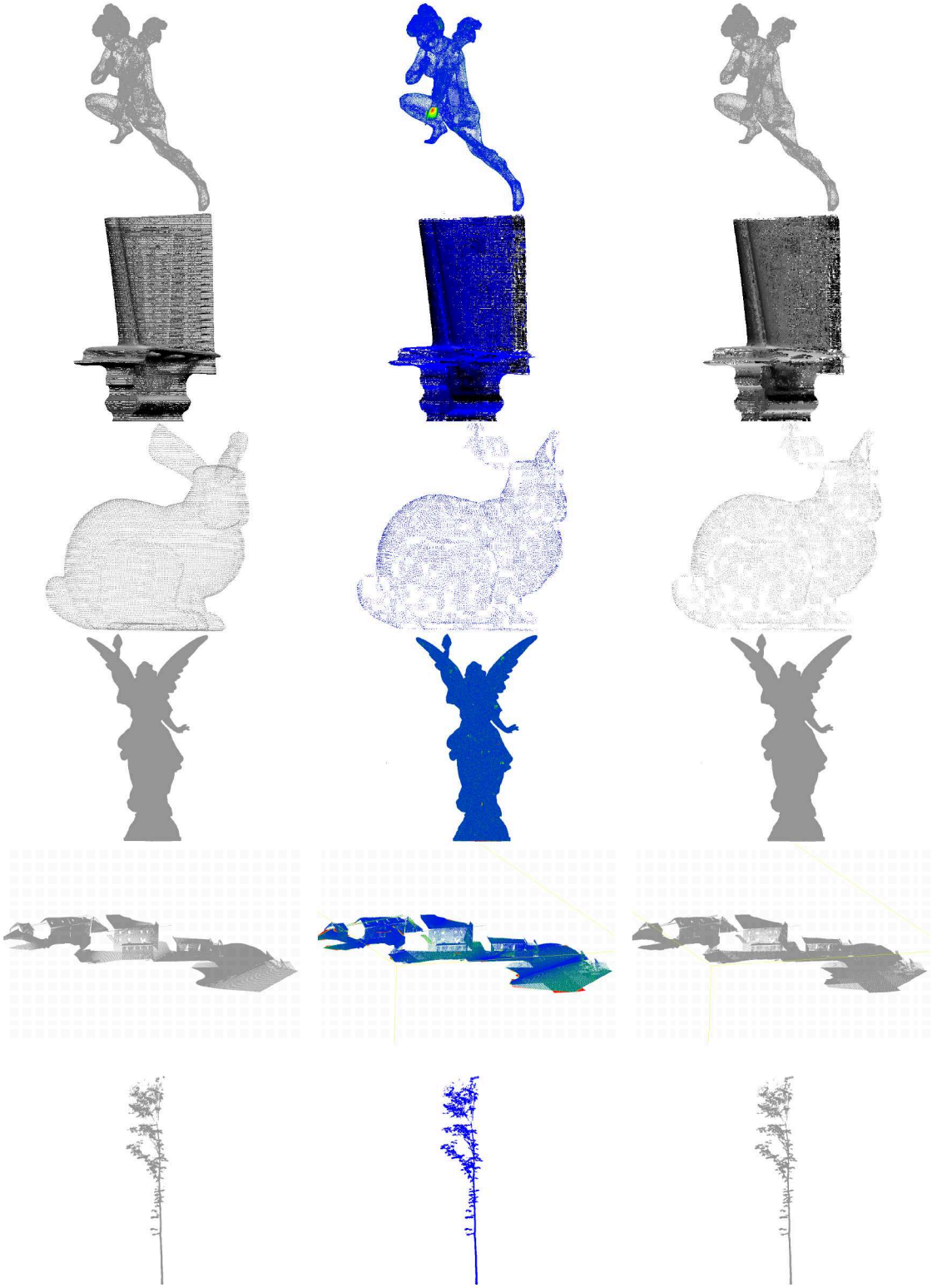


Figure III.11. – Results of the compression algorithm. From left to right: original point cloud ; distance map from compressed to original (red points indicates higher distances) ; compressed point cloud.

Modèle	Original	XYB	Octree	Gain
Angel	13,8 Mo	6.2 Mo	2 Mo	6.9
Blade	56,8 Mo	23 Mo	5.3 Mo	10.7
Bunny	2,1 Mo	935 ko	301 ko	7
Lucy	884,6 Mo	364,7 Mo	43,3 Mo	20.4
Tree	57,2 Mo	25,2 Mo	3,4 Mo	16.8
Urban	40 Mo	16,7 Mo	1,4 Mo	28.6

Table III.2. – Quantitative evaluation of the compression algorithm.

4. Discussion

We achieved our objective and developed a method for fast normal estimation. Indeed, tests have demonstrated the running time efficiency of our methods. The calculations by surface patches instead of point-wise operations reduces the complexity of the normals estimation. Whereas the implementation of our anisotropic subdivisions is mono-thread, the CloudCompare normals estimation methods are multi-thread processes taking advantage of the eight cores available on the computer used during the tests. Thus, using thread partitioning libraries could even increase the benefit in terms of running time. Besides normal directions, our methods also provide a curvature indicator and an efficient point cloud compression.

Comparing our methods on six data sets with different characteristics (number of points, sampling density, and type of model) provided sufficient content for evaluation. Our two methods differ in terms of running time efficiency and accuracy. Whereas the $RMSE$ octree performs slower than the σ_3 octree, it generates errors in the range of the Meshlab and CloudCompare methods. The σ_3 octree subdivision, however, produces larger errors, mainly due to its inability to capture details and sharp features. Nonetheless, the σ_3 octree reached the same accuracy as other methods when it was applied on a point cloud with a majority of flat surfaces. Furthermore, patches identified as *edge* cannot be modelled by a surface. Consequently, normals are not estimated on such points, which, in turn, impacts the surface orientation along neighbouring patches.

The impact of noise on the resulting normals has not been studied through an extensive validation yet and is the next step towards a deeper understanding of the capacities of the proposed normal estimation. Robustness to noise was not the main motivation of the presented developments, rather we focused on the balance between computing time and accuracy, as stated earlier (other existing methods were developed to overcome the problem of noisy data). However, we provide some elements to analyse the potential response of our methods. First,

given a point cloud affected by a certain level of noise points, processing larger patches (larger than point-wise neighbourhoods) improves surface fitting. This in turn should improve the robustness of normal estimation (let us point out that thanks to these larger patches, we could also detect and remove outliers during surface fitting to mitigate their impact). When considering a spherical neighbourhood, the chosen radius has to be set carefully: it should be higher than the distance from noisy points to the underlying surface. Otherwise, the local dispersion of the cloud is too important to obtain accurate information. The same rule applies to the consideration of the k-closest neighbours, which generally implicitly engenders a small neighbourhood radius. By contrast, the subdivision into patches allows to analyse a noisy surface over larger areas. In such case, the noise dispersion in less likely to amalgamate noise and pertinent information. Overall, the chosen scale of analysis during the surface fitting is a critical aspect that has to be carefully considered in optimisation-based approaches.

Our work opens up several perspectives in terms of performance improvements and new applications. A way to improve the efficiency of our method is to take advantage of both the σ_3 and $RMSE$ criteria to obtain a compromise between efficiency and estimation accuracy. We believe this could be achieved by using a criterion based on the value of λ_3 rather than the σ_3 indicator. We also intend to explore the relevance of trilinear interpolation during normals blending and apply it to estimate normals on currently ignored patches. Octree structures are a major element in computer graphics and they take part in many algorithms in order to accelerate the processing. KD-trees are also a popular space partitioning structure. We chose to use octrees because of the regularity of the subdivision (the space is divided into cubes) and because of the efficiency of octrees for neighbourhood searches ; both of these properties are further required in other workflows. However, our method could be adapted to KD-trees with few changes. This might even enhance the anisotropy of the resulting procedure since patches could be split at regions of maximum estimated curvature. Doing so would be highly relevant to obtain the larger uniform patches. Another improvement would be to develop these changes in a multi-thread implementation. Additionally, the method could be adapted to handle out-of-core point clouds containing up to several tens of billions of points. Indeed, once patches have been identified and approximated with a quadratic surface, the data points are no longer needed. Thus, the only memory footprint remaining is that of the octree structure itself, which is manageable. Therefore, loading separately appropriate chunks of the point cloud is a viable solution that do not interfere with the normal orientation procedure. Moreover, out-of-core clouds usually represent wide areas (e.g. an entire and detailed building, or parts of a city). With these data, it would be recommended to impose a maximum patch size prior to the octree creation. For example, patches of 15 m^2 would have a great chance of containing sharp features with a junctions of different surfaces such as wall-

ground. This patch size limit would fasten the procedure by avoiding several subdivisions and make it easier to handle the amount of data points. Finally, since anisotropic octrees offer a hierarchical representation of the data (from coarse to fine) they could be used to efficiently store and generate various level of details of a given point cloud.

5. Conclusion

We have presented two new methods to estimate normal directions and curvatures on a point cloud which are major elements in geometric analysis. Both rely on an efficient anisotropic octree subdivision of the data and perform faster than the benchmark pieces of software tested. Results show that the $RMSE$ criterion is preferable for capturing great details on the point cloud, whereas the usage of the faster σ_3 criterion has to be restricted to point clouds containing mostly flat surfaces (e.g. urban or indoor environments).

The generated anisotropic octrees represent a multi-purpose and efficient tool for point cloud handling. Indeed, besides the normal and curvature estimation presented here, octrees are included in a wide range of applications in computer graphics. As an example they are almost inevitable for visualisation and neighbours finding, and may be involved in segmentation, classification, and spatial analysis. Overall, the proposed methodology can be inserted in various workflows with few modifications and gives access to elaborated methods based on normals directions.

IV. Tubular shape extraction (STEP algotihm)

Overview

In spite of their accuracy, point clouds acquired with remote sensing technologies face common constraints such as non homogeneous sampling, occlusion and noise. Furthermore, these constraints are amplified in natural environments. Revealing the geometry of the acquired surfaces on such data sets is a challenging task and the results of existing methods are affected by these constraints. Therefore, robust point cloud processing are still required to segment, model and reconstruct objects of interest from a raw point clouds prior building any higher-level knowledge. A wide range of applications aims at extracting a specific object from a point cloud for studying and measurement purposes. This pattern recognition procedure thus requires to identify and extract a predetermined geometry from a data set. For example, extraction of tubular shapes from point clouds is of major importance. Indeed a large set of real-world objects are composed of tubular shapes, such as pipes, poles, stems or monument pillars.

In this chapter we address this issue and introduce a novel and robust method for extracting and reconstructing tubular shapes in dense, noisy, occluded and unorganized point clouds. The proposed method deals with the acquisition artefacts of point clouds from terrestrial LiDAR. We used parametric models to identify and reconstruct tubular surfaces from a data set: extracted tubular shapes $S(t, \theta)$ were parametrized by a parameter t varying along a 4D curve, and an angle θ for the revolution around the curve. These models were obtained from a combination of an original Hough transform and a new form of growing active contours (also referred to as "snakes") that overcome point clouds limitations while being able to handle large data sets. In the following we also demonstrate the robustness of our method by presenting a sensibility analysis along with applications to forest data.

1. Introduction

Extraction of tubular shapes from point clouds is of major importance, since they can be used to monitor, among others, factory constructions, refinery pipelines, or power plant structures. Tubular shape extraction is also required in forestry since tree cross-sections are commonly modelled by a circular shape. The precise reconstruction of trees and derived measurements have many applications ranging from ecology (allometric relationships, growth modelling, carbon storage assessment) to forestry (forest monitoring, sustainable development) or industry (harvests planning, sawmill optimisation).

During this thesis we developed a novel algorithm designed to extract tubular shapes from noisy and occluded dense point clouds. Besides the algorithm itself, our work incorporates two main innovations. Firstly, it introduces a novel and fast cylinder variant of the Hough transform (HT) which uses the normal vectors to lower the complexity of the classical HT. Secondly, it includes the curve parametrisation in generalised open active contour models. The proposed methodology is intended to be robust to noise and occlusions and aims at reconstructing each tubular shape present in the initial data as a separate item. We tested the tubular shape reconstruction capabilities, including a noise and occlusion sensitivity analysis, on an abstract object and on forest trees, which is our main investigation field.

Terrestrial laser scanning (TLS) acquisitions in natural forest environments include additional constraints when compared to those made in urban, or industrial point clouds. These constraints are tied to the remote sensing technique and to the complexity of the natural forest environments. TLS point cloud sampling rate may vary from one set-up to another, and the spherical geometry of the sensor results in a non-homogeneous sampling density. In addition, occlusions are particularly important in forest environments. The TLS geometry paired with the presence of vegetation (branches and leaves), induce occluded areas expanding both in size and number with increasing distance from the sensor. Moreover, noise brings supplementary confusion at surfaces extremities and is severely present in foliages. With these constraints, it is frequent that a given stem is probed with different conditions from its base up to the location where branching dominates. Forest measurements from TLS data also suffer from objects specific limitations. Even though tree stems can be assumed to have almost circular cross sections, they often deviate from this hypothesis. The non-trivial topology of the woody parts of the trees together with intricate occlusions of numerous branches complicate the point clouds processing. Furthermore, stem's bark can be irregular and rough, and hence produces disturbed surfaces. In addition, TLS point clouds of trees may be affected by wind and multiple scan alignment issues.

Therefore, several artefacts induce point cloud distortions and generate crooked objects. With these issues specific to the natural environments, forest measurements from TLS data have to overcome more constraints than those present in urban or industrial settings.

After presenting previous works in section 2 and the theoretical background of our work in section 3, we introduce our approach in section 4. In section 5 we present a validation of our approach, including a sensitivity study, both for synthetic data and terrestrial LiDAR data acquired in forest environment. These results are discussed in section 6 before we conclude in section 7.

2. Related works

Different recent approaches for tubular shapes detection are promising for the delineation of tree stems from TLS data. Among them, the so-called "arterial snakes", introduced by Li et al. in [111] are designed to detect such shapes. After computing a longitudinal vector field (locally orthogonal to normals), snakelets are initialised after clustering nearby points with similar reliable directions. They are then competitively grown, eventually merged to handle the object topology, and regularized to provide the final results. However, the clustering step for initialising snakelets may face limitations when confronted with a conical shape. In such case the normals to the clustered points cannot provide an orthogonal longitudinal direction, and thus no snakelets can be initialised. Also, the "arterial snakes" approach requires heavy computations, implying point neighbourhood search for euclidean clustering and snakes skin matching, and solving large linear systems (processing a 100k point cloud requires approximately 5 minutes). Therefore, in spite of its similarity with our work, this approach is not applicable in the context of tubular shape extraction in forest environments with major occlusions and noise, and where a single tree is sampled with several hundreds of thousands points.

Tubular structures have also been studied in [130] from an image sequence. Their method first detects and filters the junctions of the tubular structures. Then the final geometry is reconstructed using a sweeping circle along the skeleton together with a rod simulation. However, this approach requires a volumetric representation of the object, which can not be derived from single-side scanned point clouds.

Other examples of methods detecting tubular shapes were developed in the medical field. Particularly, the approach proposed by H. Li et A. Yuzzi [112], is

based on the observation that tubular shapes can be modelled by continuous curves in a 4D space. From this assumption, they use a minimal path technique to extract vessels skeletons and surfaces. However, such minimal path techniques include volumetric integration that apply only to images. As a consequence, this approach, as many works dedicated to medical data, is not applicable in the context of unstructured point clouds.

As yet, another approach integrating the normal vectors information straightforwardly has also been explored [96]. This approach accumulates information along the normals in the original 3D space in which maxima correspond to convergence voxels. Hence the result is a pure skeleton curve rather than a tubular surface. The approach only considers fixed-radius objects and is sensitive to noise, occlusions and shape variations, which is prohibitive in our context. Another skeletal curve extraction method involving normal vectors is proposed in [191]. Similarly to the study discussed above, this method does not reconstruct tubular shapes since radii are not estimated. Furthermore its complexity has to be evaluated since for each point of the input cloud, several potentially costly computations are required, including plane cutting and graph creation. Finally, the presented results illustrate that the final skeletal curve is not necessarily located at the centre of the shape. Thus reconstructing an accurate tube would require additional computations.

The forest point clouds require dedicated approaches to deal with specific constraints. Algorithms adapted to forest environment reconstruction have been proposed. They can be divided into two major classes: a "knowledge-driven modelling" class and a "geometry-driven modelling" class. Algorithms from the "knowledge-driven modelling" class circumvent the issue of geometric complexity by producing visually appealing and realistic "synthetic" trees. The tree reconstruction in these algorithms is initiated with data points and refined with botanical or forest knowledge (e.g. allometry, vegetation self-similarity, pipe model, etc) but with no shape approximation guarantee (see [121] or [208] for instance). The "geometry-driven modelling" class includes algorithms whose objective is to produce an accurate description of the trees based solely on data points and thus provide accurate forest information. These algorithms build models approximating the data with few assumptions on the shape of the objects to reconstruct.

"Knowledge-driven modelling" algorithms usually rely on a reconstruction of the tree skeleton and topology followed by a simulation entailing allometric relationships and/or L-systems [a]. First, a weighted adjacency graph is created from the data points and a defined neighbourhood. Then the tree skeleton is estimated

a. L-systems are grammars that can be adapted to generate models of plant structures according to a set of generative rules.

using either shortest path computations with Dijkstra's algorithm, or minimum spanning tree extractions from *a priori* estimated root points for each tree. Finally, the geometry of the stems is predicted starting from its skeleton paired with allometric relationships to estimate stem and branches radii. Missing parts or foliage may be added based on self-similarity or L-Systems. Therefore, resulting reconstructions are designed to be visually pleasing and close to the data points, but neither approximation properties nor precise measurements are actually guaranteed. Hence, such approaches are generally dedicated to rendering applications more than to forest monitoring.

"Geometry-driven modelling" algorithms aim at reconstructing trees by simultaneously approximating closely the data points and segmenting the woody parts of trees. Estimation of the diameter at breast height (DBH, 1.30 m above the ground surface) is essential for allometric relationships (see [47, 114, 164]). However, reconstruction of tree taper is more complex. Several algorithms have been designed to process TLS point clouds to estimate DBH and stem taper under the assumption that tree stem cross-sections can be approximated by a circle. This assumption leads to methods using shape fitting and pattern recognition algorithms. Shape fitting requires a segmentation of the point cloud into clusters to identify the points that must be fitted. For example, clusters can be produced by considering the distance from a point to its neighbour or to the cluster [33, 102], by employing a variant of k-means clustering [32], by using a structural element [21], by dividing the point cloud into patches that can then be further merged [169], or within 2D rasters [151]. Even approaches based on iterative cylinder fitting perform such clustering since points that are close to a shifted cylinder are used as a cluster for the next shape fitting step [161]. Otherwise, density-based spatial clustering may be adopted. Identified clusters are then used to fit one of the following shapes for tree stem reconstruction:

Circles can be fitted into horizontal layers [102, 152]. However, inclined stems or branches can no longer be estimated accurately since their horizontal cross-sections cannot be approximated by a circle. Ellipse fitting was also tested, but this approach faces some limitations [15, 154].

Cylinders takes into account the local orientation of tree branches. Iterative methods are capable of reconstructing entire trees from an appropriate starting point. RANSAC algorithm or principal component analysis (PCA) have been used to support cylinder fitting [151].

Cones can be used to consider stem tapering. However they are not widely used for entire stem reconstruction and produce results that are similar to cylinder fitting [95, 131].

Other shapes have also been used to achieve more precise results. For instance, B-spline fitting on horizontal layers tends to precisely match the

stem shape, but its use is limited when having to describe a single-scanned tree stem [160]. Cross-sectional polygons can also be used [79].

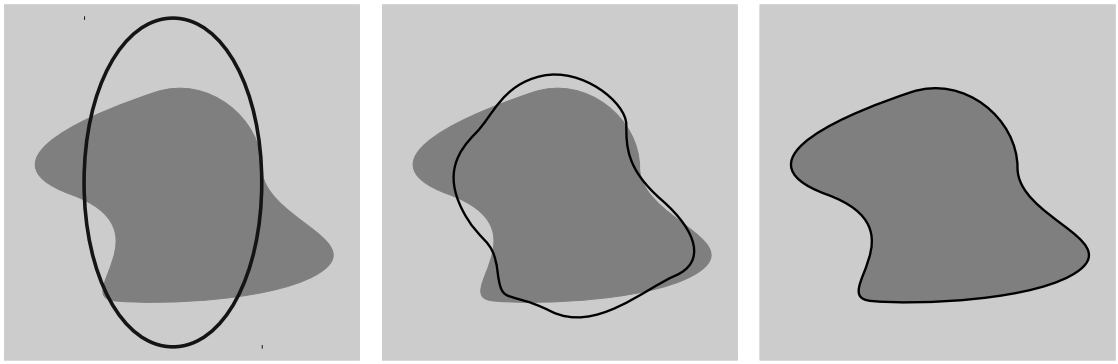
Shape fitting approaches encounter limitations. For example, a decision is required to accept or reject the fitted shape. It is usually based on a threshold that is set on the RMSE of the shape fitting. This threshold has to be set carefully and may vary from one data set to another. Moreover, least-squares fitting is influenced by noise, and its robustness to occlusion depends upon the quality of prior clustering operations. Shape fitting has been successfully applied to data sets that have been acquired under favourable conditions, but their performances may decline when the technique is applied to complex forest scenes.

Aside from shape fitting, the HT has been adapted to identify circles in order to estimate DBH or pre-locate trees [151, 180, 186]. The classical HT is not a predominant approach, mainly because of its algorithmic complexity and its high requirements for computer resources. In addition, the analysis of the HT result can be a complex task which involves several empirical criteria and thresholds. Therefore, the HT is generally reduced to a pre-localisation step. Nevertheless, the HT is attractive despite these drawbacks since it has the potential to deal efficiently with current constraints on TLS point clouds acquired in complex forest scenes.

The main objective of our study is to propose an automatic algorithm for tubular shapes reconstruction that handles the main constraints of TLS data in forest environments. Confronted to an object containing a unique connected tubular part, our method is intended to produce a single tube, whereas a shape combining several intersecting tubular parts will result in the reconstruction of an equal number of disconnected tuboids. Two methodological choices precluded the selection of solutions to deal with TLS constraints. First, we took advantage of the benefits of the HT and reduced the complexity of its computation. Thus, one of the main drawbacks of the classical HT was minimised. Second, we reconstructed each tree stem as a single entity by developing generalized open growing contours for the 4D Hough space. By doing so, we expected greater control on the smoothness and coherence of stem reconstructions.

3. Background on open active contours

Our approach is based on an original HT combined with generalized open active contours. These two notions are essential for a clear understanding of our method. Therefore, while we detailed the the Hough transform in chapter II, let



(a) (b) (c)

Figure IV.1. – Active contour deform through time to minimise their energy: initial (a), intermediate (b) and final state (c).

us now define the active contour models.

An open active contour, also called snake, is an open parametric curve $c(u)$ embedded in a discrete space (traditionally called image) and minimising an associated global energy E_g [92] (Figure IV.1). The energy is defined such that it reaches its minimum when the curve fulfils desired properties. Minimising this energy results in a compromise between different constraints that are expressed through its definition. Active contours were initially introduced by Kass et al. in [92] and further developed [42, 204, 207]. It classically uses the following energy:

$$E_g = \int \Big[E_i\big(c(u)\big) + E_d\big(c(u)\big) + E_e\big(c(u)\big) \Big] \, \mathrm{d}u \tag{IV.1}$$

where $E_i\left(c(u)\right) = \alpha|c'(u)|^2 + \beta|c''(u)|^2$ is related to the internal geometry of the curve, $E_d\big(c(u)\big)$ is a data-related term that depends upon the value of the image, and $E_e\big(c(t)\big)$ is a more general term that includes external (i.e. user defined) additional constraints, such as local repulsive forces. Specifically E_i aims at controlling the elasticity and curvature by applying a constraint upon the first and second derivatives of the curve, and E_d constraints the curve to evolve toward elements of interest in the data image.

The computation of active contours relies on the minimisation of E_g, and thus, on a multi-variable optimisation scheme. This optimisation requires, in turn, an initial guess of the location of the curve in the neighbourhood of its optimal state. Using Euler-Lagrange equations, the minimisation is first transformed into a partial differential equation, and then solved iteratively using Euler schemes

through time t:

$$c(t) = (\mathrm{A} + \gamma \mathrm{I})^{-1} \left(\gamma c(t-1) - \nabla E_d(t-1)\right) \tag{IV.2}$$

where A is a pentadiagonal banded matrix used to approximate the derivatives of the curve and γ a time step. The resulting curve is a compromise between different constraints: mainly, the curve geometry and its position in the image. Thus, it represents a powerful and attractive tool for extracting a desired smooth curve in possibly noisy data.

4. Methodology

Let us define a tuboid as an ordinated series of 3D circles with continuous locations, orientations and radii. It is equivalent to the envelope of a continuous series of spheres. Our algorithm extracts such tuboids from dense point clouds. Actually, in addition to the point cloud, our approach requires a normal vector field which can be either provided with the point cloud (according to the scanning technology used) or computed. Such normals computation has been widely studied and we refer the reader to [99] for a survey. Our method involves two main components (Figure IV.2): (1) defining an original HT to identify efficiently 3D circles (or equivalently spheres) in raw point clouds. A previous work by Kimme et al. [97] bears some similarities but identifies 2D circles in 2D images. And (2) defining generalized growing open active contours within HS to identify and link the most representative 3D circles, thereby forming a fully coherent tuboid via energy minimisation. Finally, we transform each active contour back into tuboids in the original Cartesian space. This procedure identifies a set of tuboids within a global scene. In the context of point clouds from TLS acquisitions in forest area, each tuboid represents a tree stem with direct access to DBH and taper. In the sequel, we will refer to our approach as Snakes for Tuboid Extraction from Point clouds (STEP).

4.1. Point-normal circles Hough transform

The first part of the STEP method is to define and design a computationally efficient HT to identify potential 3D circles (or equivalently spheres). The resulting HS will then provide a reliable space for initialising growing open active contours

from local maxima. Our HT relies on three main elements. First, normal directions of the points are used as additional information to reduce the complexity of the HT calculation. Second, a filter is applied in the HS to discard elements of low interest, thereby reducing the complexity of the subsequent space analysis. Third, maximal HS elements are selected as the best candidates for extracting circular cross-sections, and further used as seeds for active contour growing.

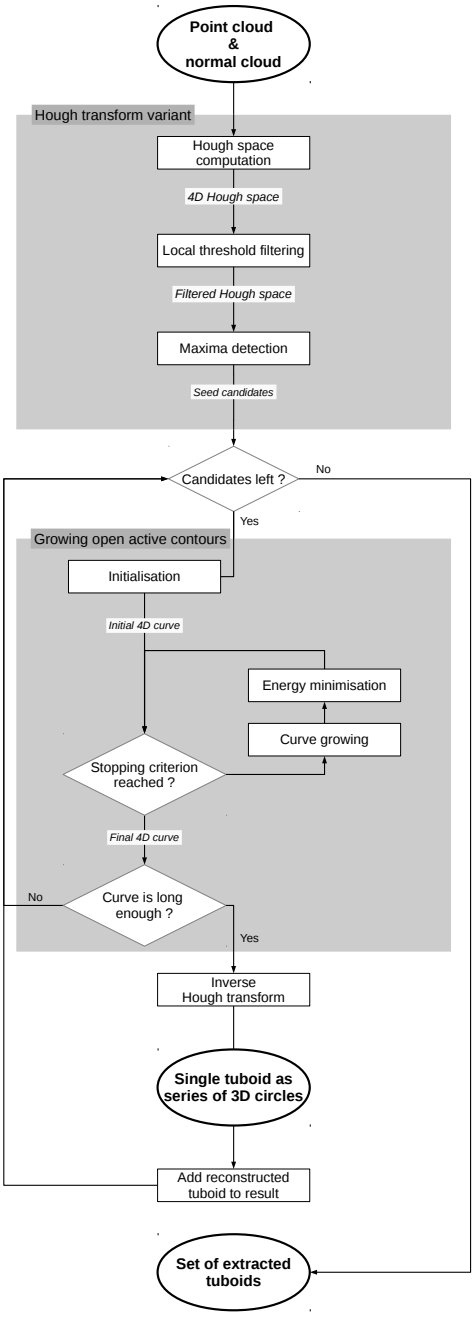
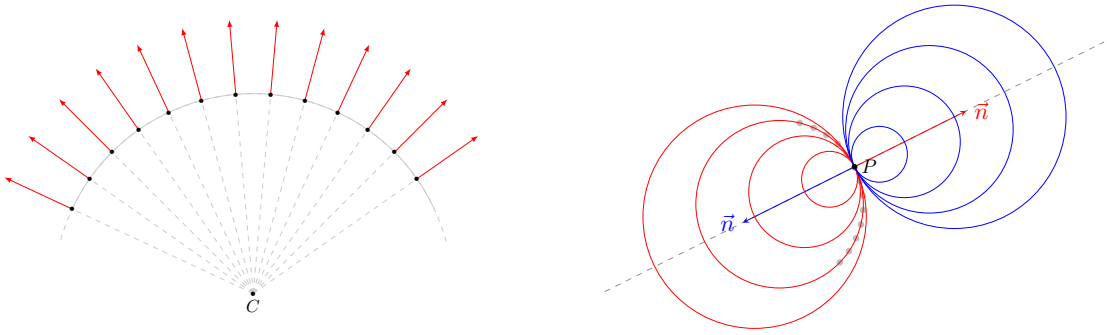


Figure IV.2. – Overview of the STEP methodology.



(a) (b)

Figure IV.3. – Normal convergence properties: (a) Opposite normal directions converge towards the center of a circle. (b) A point p with normal $\vec{n}$ votes for a set of circles (red). Considering $-\vec{n}$ involves a second set of circles (blue).

4.1.1. Hough space computation

We intend to tailor the classical HT to detect 3D circles. In a straightforward approach, a circle is represented by 7 parameters: $C = (\vec{c}, \vec{n}, r)$ where $\vec{c} \in R^3$ is the location of the centre of the circle, $\vec{n} \in R^3$ its normal direction and r its radius. Such settings would therefore classically lead to the computation of a 7D discrete HS, which is both time consuming and memory costly. However, in a tuboid, orientations of circles can be retrieved later from the skeleton. Therefore, we reduce the number of dimensions of the HS to 4, namely $(\vec{c}, r)$. Doing so, we take advantage of the HT while reducing the space complexity of the HS.

Our HT also reduces the algorithmic complexity of the voting phase by considering the information provided by normal directions, in addition to points location. We start from the following property of circles: given a point p on a circle C and its normal $\vec{n}_p$ (oriented outward), the half-line defined by p and $-\vec{n}_p$ passes through the centre of C. That is, the opposite normal vectors of the points on a circle converge towards its centre (Figure IV.3a). Therefore, the set of circles containing $(p, \vec{n}_p)$ is the set of circles $C = (\vec{c}, r)$ of centre $\vec{c}$ and radius r satisfying the following system of equations, which is the parametric equation of a half line:

$$\begin{cases} c = p - \lambda \vec{n} \\ r = \lambda \end{cases}, \ \forall \lambda \in \mathbb{R}^+ \tag{IV.3}$$

However, estimating a coherent vector field of outward-oriented normal vectors on a point cloud is a challenging issue. When there is no possibility of choosing a consistent orientation, both directions have to be considered and points

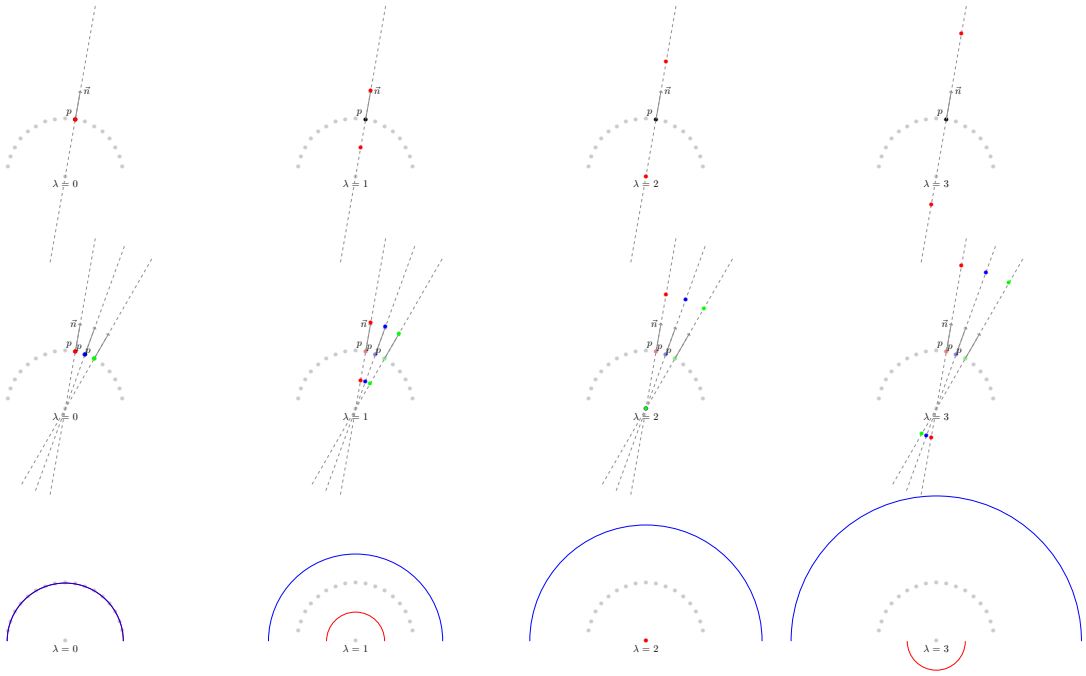


Figure IV.4. – Properties of the computed Hough space (represented in 2D) on a point cloud (grey). Top row: the two votes of a data point p along its normal vector for different values of λ. Middle row: the votes of three data points for different values of λ. Bottom row: when the votes are extrapolated in a continuous domain, two distinguishable sets of vote appear corresponding to inner circles (red) and outer circles (blue).

then vote for two half lines in the 4D HS (Figure IV.3b):

$$\begin{cases} c = p \pm \lambda \vec{n} \\ r = \lambda \end{cases} , \forall \lambda \in \mathbb{R}^+ \tag{IV.4}$$

Hence, depending on whether the normals are oriented or not, Equations (IV.3) or (IV.4) are used to accumulate the votes of the data points. These votes can be computed efficiently in linear time (with respect to the HS resolution) using fast-ray tracing algorithms (such as [5]) adapted to 4D spaces.

4.1.2. Hough space filtering

As stated previously, when normals are not oriented, data points vote for two half-lines in the Hough space: one half line correspond to inner circles, and the other to outer circles (respectively coloured in red and blue in Figure IV.3b). Fi-

gure IV.4 illustrates this phenomenon in the Hough space. It is obvious that outer circles are not representative of the point cloud and have to be ignored during further processing. Filtering these elements from the Hough space would not only facilitate further analyses, but also reduce computing time. We developed two different filters to achieve this goal. The first one is based on a theoretical analysis of the Hough space. It gives accurate results but is time consuming. On the contrary, the second filter is based on a greedy heuristic and performs faster.

Theoretical filter

Figure IV.4 reveals how the votes in the Hough space are distributed over two classes: inner circles and outer circles. When data points are regularly sampled, the score inside each class tends to be homogeneous. However, the score of inner circles is significantly higher since normal vectors converge toward inner circles and diverge toward outer circles. We used this property to classify the Hough space elements and filter outer circles from the Hough space with Otsu's threshold [153]: this threshold is well suited for fast binary classifications because is defined to maximise inter-classes variance. However, this operation must take into account the properties of TLS point clouds.

The score corresponding to a given circle depends on its radius and on the local sampling density. Indeed, assuming a constant point density on a point cloud, a large circle collects more votes than a small one. Under the non-homogeneous sampling density due to TLS, two circles of equivalent radius also have different scores. Hence it is not possible to perform a global classification of the Hough space elements. We rather apply the binary classification in a local analysis window for each discretized radius r in the Hough space. We set the width of this window to $\alpha 2r$ ($\alpha \geq 1$), which is greater than the distance separating inner scores from outer scores. We also ensure that the two classes are present in the analysis window by analysing the ratio of the average score in each class and comparing it to a theoretical value. Considering n voting points located on a circle, the averages score a_1 and a_2 of the inner and outer circles are computed as (Figure IV.5):

$$\begin{aligned} a_1 &= \frac{n}{l1} = \frac{n}{\varepsilon\alpha} \\ a_2 &= \frac{n}{l2} = \frac{n}{(2r+\varepsilon)\,\alpha} \end{aligned} \tag{IV.5}$$

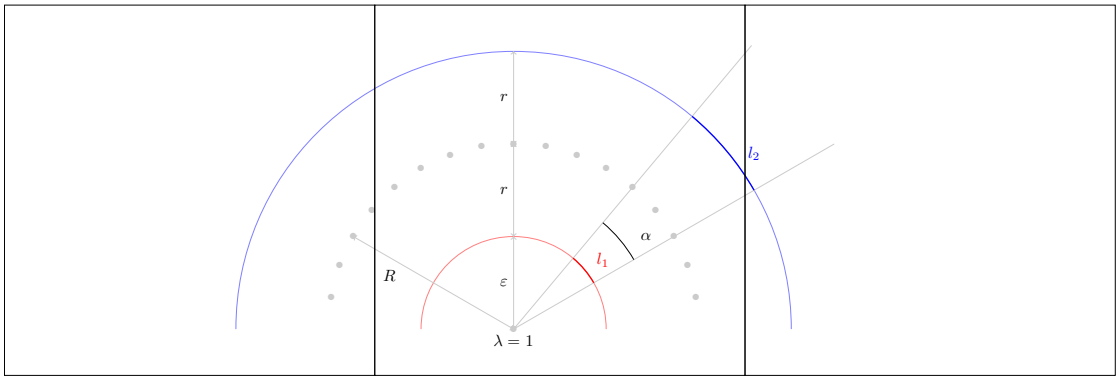


Figure IV.5. – Local filtering of the Hough space using analysis windows. The left and right windows do not include the two classes of scores.

The ratio:

$$\Delta = \frac{a_1}{a_2} = \frac{\varepsilon}{\varepsilon + 2r} \tag{IV.6}$$

only depends on the radius r of the Hough space being analysed and can be estimated theoretically. Therefore, if the ratio of two classes derived by Otsu's threshold differs from the theoretical value, we considered that only outer circles were included in the window analysis, and we filtered the entire elements inside this window.

The proposed filter has a strong theoretical background based on the properties of the Hough space. However, in practice the number of analysis windows required to filter the entire Hough space which makes the filter computationally expensive.

Empirical filter

In order to avoid the time consuming theoretical filter presented above we introduce an alternative fast filter. We directly detect the accumulation of higher votes along the half line corresponding to inner circles, and filtered Hough space elements traversed by the half line corresponding to outer circles. To do so, we repeated the raytracing procedure used during the voting step for each data point. The final filter was achieved by computing the sums of scores s_1 and s_2 along each half line, and filtering the elements of the Hough space along the half line with the minimum sum of score.

Note that $s_1 \simeq s_2$ when score accumulation is not observable (i.e. in presence

of non circular elements). Hence both half lines were filtered when:

$$\frac{max(s_1, s_2)}{min(s_1, s_2)} < \tau \tag{IV.7}$$

where τ is a threshold set y the user. In our applications τ was empirically set to 2.

4.1.3. Maxima detection

The resulting HS is a discrete 4D image in which each element represents an occurrence of a 3D circle. The value of each element within this space is the voting score of the corresponding circle. The HS must be analysed to extract circles of interest: elements with the highest score are most representative of the data. Yet, the score itself cannot be considered as the only reliable criterion for circle extraction: two effects alter the number of points that are sampled on each circle. First, point sampling rate varies with the distance to the sensor. Stem cross-sections with similar radii can then be described by different number of points. Second, even if constant sampling is assumed, circles of smaller radius will be sampled by a fewer points. Under such conditions, circles of interest cannot be extracted with a given threshold or solely with a local maxima detection over the HS, as can be done in other applications. Rather, in a first step we extract local maxima (considering a direct 4D neighbourhood), regardless of their scores, and use them as seeds for growing active contours.

4.2. Growing open active contours

Our second step is to analyse the HS to identify sets of tuboids. Tuboids are open curves in the Hough space, hence we need to extract smooth curves that pass through the local highest scores of the HS. Therefore, growing open active contours emerge as a well suited extraction approach. However, we noticed unwanted interaction between the classical snake energy and the growth process: energy minimisation actually hinders the growth. We define a new energy for active contours which prevents such interactions by taking parametrisation into account.

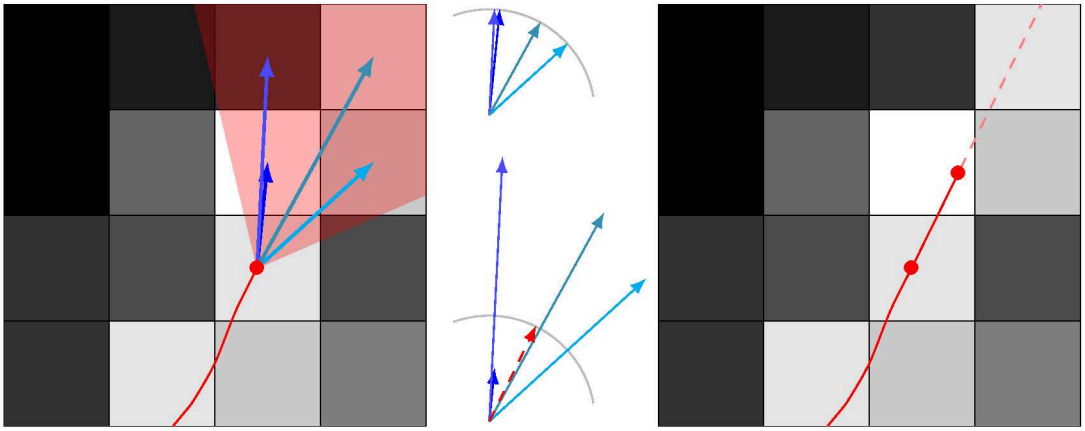


Figure IV.6. – HS represented as a 2D image with intensity related to the HS score. Left: potential growing directions within a neighbourhood (light red) cone overlaid with an active contour (red). Centre: directions are normalised (top) and scaled according to their score (bottom), while the final growing direction (light red) is found by UPCA. Right: a point is added in this direction from the contour's extremity.

4.2.1. Energy definition and minimisation

In the context of our study, we intend to extract (1) smooth curves (i.e. smooth tuboids in terms of location, orientation and radius), and (2) curves passing through elements with high score in the HS. We define a new global energy $E_g\big(c(u)\big)$ expressing both of these constraints for the open active contour:

$$E_g\big(c(u)\big) = \int \Big[E_i\big(c(u)\big) + E_d\big(c(u)\big)\ \|c'(u)\|\Big]\ \mathrm{d}u \tag{IV.8}$$

Actually, most active contour approaches are used in the context of images or binary volumes. Hence classical snakes are based on pixels and implicitly assume curvilinear parametrisation of the curve. However, we model snakes as continuous, piecewise linear curves deforming over time. Therefore, unlike classical energy (Eq. IV.1), the term $E_d\big(c(u)\big)\ \|c'(u)\|$ computes the integral of the data energy along the curve (given in general parametrisation). We thus obtain a parametrisation-independent formulation of the data energy. This consideration entails important changes in the minimisation scheme and provides a stable growth of the curves.

Let us now more precisely define $E_d\big(c(u)\big)$. We denote $H\big(u\big) = HS\big(c(u)\big)$ as the value of the HS at $c(u)$, H_m and H_M the respective minimum and maximum values of the HS, and $h_m(u)$ and $h_M(u)$ the respective minimum and maximum

values of HS in the neighbourhood of $c(u)$. We define data energy as follows:

$$E_d\Big(c(u)\Big) = a\frac{H_m - H\Big(u\Big)}{H_M - H_m} + (1-a)\frac{h_m\Big(u\Big) - H\Big(u\Big)}{h_M\Big(u\Big) - h_m\Big(u\Big)} \tag{IV.9}$$

It is the weighted sum of a global and a local term with $a \in [0,1]$ a balancing variable. The global term normalises HS scores over the entire space. However, such normalisation does not take into account local score variations. The local term does account for local variations, but induces high energy variation in small areas. The proposed data energy combines both expressions above to obtain a regular energy value over the HS, while preserving the importance of the local score.

Our global energy is the integral of a functional F:

$$E_g = \int F\Big(u, c(u), c'(u), c''(u)\Big)\ \mathrm{d}u \tag{IV.10}$$

From Euler-Lagrange equations, Equation (IV.10) reaches its minimum when:

$$\frac{\partial F}{\partial c(u)} - \frac{\mathrm{d}}{\mathrm{d}u}\left(\frac{\partial F}{\partial c'(u)}\right) + \frac{\mathrm{d}}{\mathrm{d}u^2}\left(\frac{\partial F}{\partial c''(u)}\right) = 0 \tag{IV.11}$$

Developing Equation (IV.11) for the energy given in Equation (IV.8) leads to:

$$-\,2\alpha\ c''(u) + 2\beta c''''(u) + v_1 - wv_2 = 0 \tag{IV.12}$$

with $w = \dfrac{E_d(c(u))}{\|c'(t)\|^2}$, and where v_1 and v_2 are matrices which lines are given by $v_1(u)$ and $v_2(u)$ defined as follows:

$$v_1(u) = \|c'(u)\|\nabla E_d(c(u)) - \frac{\langle\ \nabla E_d\left(c(u)\right)\ ,\ c'(u)\ \rangle}{\|c'(u)\|}c'(u) \tag{IV.13}$$

$$v_2(u) = \|c'(u)\|c''(u) - \frac{\langle c''(u)\ ,\ c'(u)\rangle}{\|c'(u)\|}c'(u) \tag{IV.14}$$

Interestingly enough, $v_1(u)$ is the component of the data energy gradient orthogonal to the curve tangent, and $v_2(u)$ is the component of the second derivative orthogonal to the curve tangent. Discretising the first and second derivatives, Equation (IV.12) becomes:

$$\mathrm{A}c + v_1 - wv_2 = 0 \tag{IV.15}$$

Following the approach introduced by [92], we then consider the active contours as dynamic systems through time t, the steady state of which is given by Equation

(IV.15). We solve the resulting partial differential equation using a combination of implicit and explicit Euler schemes. Thus a solution is found using the following iterative scheme:

$$c(t) = (\mathrm{A} + \gamma \mathrm{I})^{-1} \left(\gamma c(t-1) - v_1(t-1) + w v_2(t-1)\right) \tag{IV.16}$$

where γ is a time step. Integrating the data energy along the curve without assuming curvilinear parametrisation induces a major difference with the appearance of $-v_1(t-1) + wv_2(t-1)$ replacing the gradient of the data energy term present in the original optimisation scheme (Eq. IV.2). In particular, it follows that the minimisation of the data energy only deforms the contour in directions orthogonal to the curve. This point is fundamental for growing open active contours. Indeed, snakes grow along lines of high scores ; such lines themselves contain local maxima and gradient along them can either shrink or stretch the curve with the classical snake energy formulation.

With our generalized active contours, data energy constraint does not conflict any more with the curve growth at its extremities. Therefore, our optimisation scheme has a major effect on curve evolution towards its optimal position, which is an important consideration when dealing with growing open active contours.

4.2.2. Initialisation

In point clouds, no *a priori* information is available regarding the location or the length of the tuboid that is to be reconstructed. To address the initial location issue, we take advantage of the previously extracted HS local maxima. For each local maximum e of the HS, a segment is initialised as a growing open active contour seed. Let $n_1 \cdots n_k$ be the set of neighbours of e with a respective score s_i. For any $i = 1, \cdots, k$, we consider the weighted direction $\vec{d_i} = s_i \frac{\overrightarrow{n_i - e}}{\|\overrightarrow{n_i - e}\|}$ (Figure IV.6). We then run an uncentred principal component analysis (UPCA) on the set of directions $\vec{d_i}$. The orientation of the initial segment is set to the eigenvector that is associated with the highest eigenvalue of the UPCA.

4.2.3. Curve growth and stopping criterion

Initial segments iteratively grow towards high scores in the neighbouring HS outside the minimisation scheme. The energy minimisation procedure is instead

regularly interleaved between curve growth operations and performed again as a final step. At each iteration, and for both extremities of the curve, a growing direction is computed. We avoided the growth of the curve towards the closest local maxima since noise would impact this local choice. Moreover, this might induce a rigid growth of the curve when a smooth one is preferred. Instead the computation takes into account a fixed number of the HS elements of highest score inside a cone that is centred at each ending point of the active contour. Each selected element within this cone is considered as an attractor for the curve. The potential growing directions are computed in a manner similar to that of the orientation of the initial segment, except that only neighbours inside the cone are considered for UPCA. As previously, the growing direction $\vec{G}$ is considered to be the eigenvector associated with the highest eigenvalue of the UPCA.

Growth stops when the curve reaches the boundaries of the HS or when a stopping criterion is met. In our study, the lack of preferential growth direction is used as a stopping criterion. Let us denote the eigenvalues of the UPCA by $\lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \lambda_4$. When the ratio $\sigma = \dfrac{\lambda_1}{\lambda_1+\lambda_2+\lambda_3+\lambda_4}$ is lower than a threshold, the first component of the UPCA is no longer considered sufficiently important to describe a growing direction and the growth of the curve is stopped.

Combining (1) the initial location of small curves at the local maxima of the HS, (2) curve growth along HS elements with high scores, and (3) energy minimisation solves the lack of *a priori* information on the final tuboids that are to be detected.

4.3. Elements of complexity

In order to better appreciate the advantages of our HT with respect to the classical HT, let us provide some elements on the complexity of both approaches. The classical HT can actually be either *many-to-one* (a set of points votes for a single model) or *one-to-many* (a point votes for the set of models it belongs to. The complexity of the *many-to-one* approach for circles or spheres is $\mathcal{O}(n_p^3)$, with n_p the number of data points. This is thus impracticable on point clouds containing several millions of points. The *one-to-many* approach would be more suitable. However, even in this case, for any radius r in the range of $[r_{min}, r_{max}]$ discretised into n_r bins, a point votes for a full sphere of radius r in the 3-dimensional space of centre locations. This voting procedure is thus $\mathcal{O}(n_p n_r r_{max}^3)$, which is still computationally expensive. In contrast, the normal information integrated to our method reduces this complexity. According to Equation (IV.4), each point votes for a discrete line in the HS. With a fast raytracing algorithm, the vote is

thus linear with respect to the number of bins. In turn, the global complexity of our variant of the HT is only $\mathcal{O}(n_p n_r)$. Once the HT has been computed, the active contours complexity is dominated by the energy minimisation since the growth is computed in constant time. The complexity of the energy minimisation scheme depends on the number k of samples in the active contour. The most costly operation during this procedure is the inversion of the $k \times k$ matrix A. This matrix being pentadiagonal, the complexity of this inversion is $\mathcal{O}(k^2)$.

The memory footprint of the proposed methodology depends on the implementation: whether the HS is represented as a dense or as a sparse 4D matrix. In the case of a dense matrix, assuming that an integer is coded on 4 bytes, the creation a HS of dimension d_x, d_y, d_z, d_r requires $4 \times d_x \times d_y \times d_z \times d_r$ bytes. Let us consider, for example, a $(10\ m)^3$ point cloud where the radius of the reconstructed tubes vary from 5 cm to 20 cm. The creation of a HS with a resolution of 2 cm for the x, y, and z dimension and 1 cm for the radius results in a 1.88×10^9 elements array, which is close to the maximum integer 2.15×10^9 and would require approximately 7.5 Go of main memory. In contrast, it is hard to precisely estimate the memory footprint of sparse implementations as it depends on the distribution of points. The HS footprint for reconstructing the trees presented in the results section is in the approximative range of 400Mo to 900Mo. As an additional indicator, a 15 m radius circular forest plots with trees up to 20 m height with the same resolutions as above required approximately 5.5Go in memory. Even though our method is demanding, such requirements are in the range of current hardware capacities.

5. Results and interpretation

The STEP method was tested on four different data sets. For all of them, we used the plugin available in Meshlab to estimate a coherent normal vector field for the point cloud. The first one was composed of simulated point clouds sampled over cylinders. They were used to evaluate the sensitivity of the STEP method to specific constraints of TLS data. The second data set involved simulated acquisitions of a complex tubular object (Figure IV.7). It was exploited to show the capacity of the STEP algorithm to reconstruct objects with varying geometry under different sampling conditions. The third data set was composed of *in situ* TLS acquisitions of a tree in a natural forest plot. The comparison of our stem reconstruction with high precision measurements allowed quantifying the pertinence of the STEP method in a real forest environment. The last data set was included to show the ability of our algorithm to extract the main branching structures of a natural stand using a TLS point cloud.

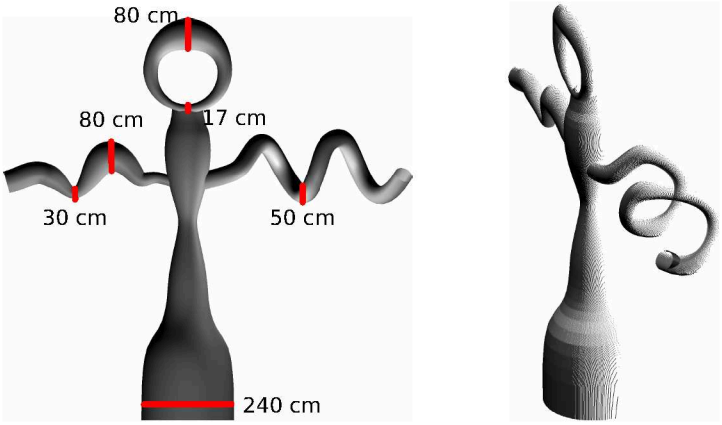


Figure IV.7. – Second data set. Left: 3D model used to simulate the point clouds, the diameter of different areas of interest are indicated. Right: simulated single-sided point cloud.

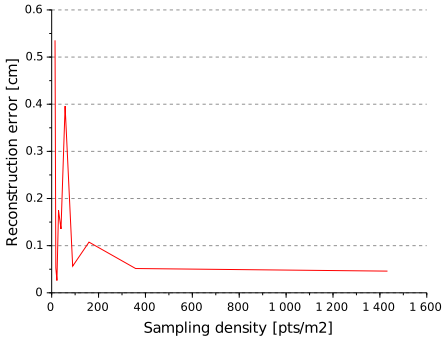
First data set

The first data set was composed of a cylinder with a fixed radius of 50 cm and a fixed length of 2 m, scanned under different conditions to exhibit the behaviour of our approach with respect to several issues: undersampling, occlusion and noise. The resolution of the corresponding HS was of 2 cm for the x, y, and z dimensions and 1 cm for the radius.

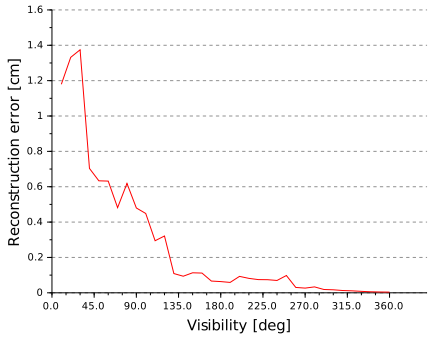
It was first scanned with ten decreasing point densities, ranging from 14 pts/cm^2 to almost 1,4000 pts/cm^2. Each simulated cylinder was reconstructed with a radius estimation error ranging from 0 to 0.75 cm, with an average error, for each cylinder, ranging from 0.026 cm to 0.54 cm and a standard deviation of 0.17 cm (Figure IV.8a). These results thus demonstrated the robustness of the STEP method to changes in point density, keeping the error on the estimates of cylinder radii generally less than 0.5 cm.

In the second test, the same cylinder was used to produce occluded point clouds. To simulate occlusion, the cylinder was sampled as if only a small portion of its surface was visible. This sampled portion of the cylinder varied from 10°(only a small part of the cylinder is sampled) to 360°(the whole cylinder is sampled) by increasing steps of 10°. The cylinders were reconstructed using the STEP algorithm, with recorded errors on the radii in the range of 0.004 cm to 1.37 cm (Figure IV.8b). Average error on the radius was 0.27 cm with a standard deviation of 0.38 cm.

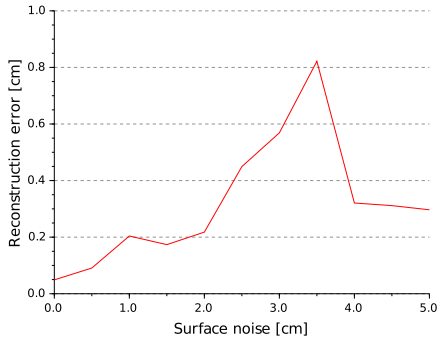
The third test evaluated the effect of surface noise on our algorithm. It was evaluated by simulating acquisitions with different amounts of random noise in



(a) Sensibility to density.



(b) Sensibility to visibility

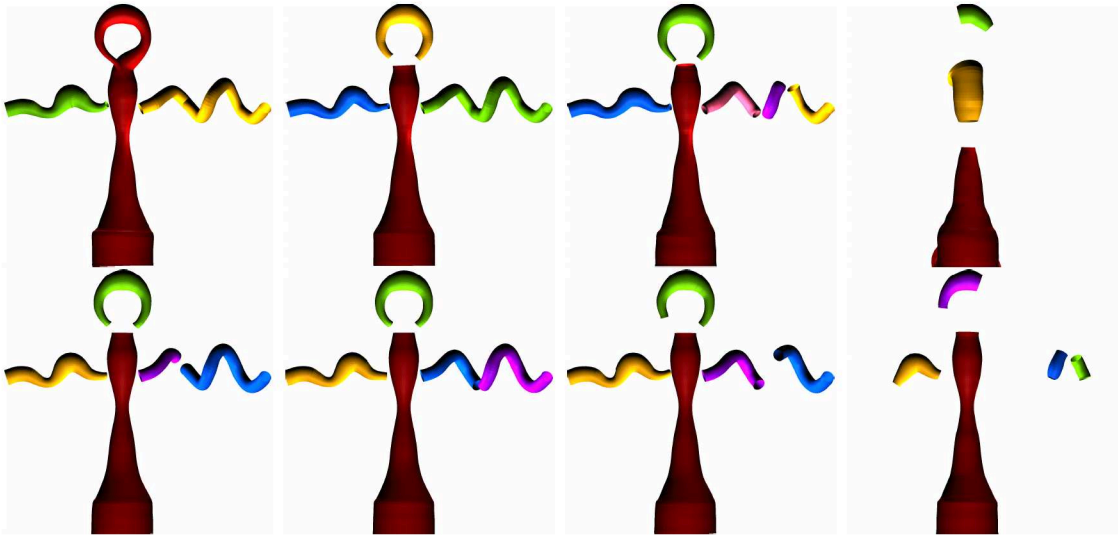


(c) Sensibility to noise

Figure IV.8. – Sensibility analysis of the STEP method on simulated cylinders

the points coordinates. The amount of noise varied from 0 cm to 5 cm (corresponding to 0% to 10% of the cylinder radius). The overall average error on stem diameter using the STEP method was 0.32 cm with a standard deviation of 0.22 cm (Figure IV.8c). Considering the amount of noise in the data set and the maximum average error of 0.82 cm, the error is well within the expected accuracy range of 1 cm.

The results obtained from these three tests set illustrate the stability of our algorithm despite the limitations of TLS data. Whatever the sampling rate, the error is within the range of the HS radius resolution used. Indeed, the property of normal convergence does not depend on sampling issues. The second tests demonstrate the resilience of the STEP method to occlusion. From (Figure IV.8b), it appears that with a sampled portion of more than 40°, the errors are below 1 cm and lessen as visibility increases. The simulations also illustrate the rise of the reconstruction error with respect to additional surface noise. With errors less than 1 cm for each level of noise, the response of the STEP algorithm is coherent with the chosen resolution. Actually, surface noise lowers the quality of



(a) 25% of missing points (b) 50% of missing points (c) 75 % of missing points (d) 90 % of missing points

Figure IV.9. – Evaluation of the robustness of the STEP method when confronted to various amounts of points. While the top and bottom rows illustrate respectively the reconstruction of the single-sided model and two-sided model, the amount of missing points is specified for each corresponding column.

normals vector computation and decreases the uniformity of the normal vector field. However, since the HT is an accumulation approach, the score convergence is still observable and the noise may compensate itself. This effect is visible on figure IV.8c where the error lessen for a higher level of surface noise while we expected the error to increase the reconstruction error.

Second data set

The second data set contains simulations of time-of-flight camera acquisitions of the geometrical model depicted in figure IV.7. With this object, we intended to analyse the behaviour of our approach on challenging features that may appear in real data such as tapering, windings, and strong radii variations.

We simulated two acquisitions approximating a common TLS resolution of 3 pts/cm^2 10 meters away from the sensor: one from the front of the object, and the other from the back. Since the model is larger than the previous cylinder, the HS resolutions were set to 3 cm for the x, y, and z dimensions and 1 cm for the radius. These two configurations allowed looking at the accuracy of the shape reconstruction for a single-sided and a two-sided point cloud of a complex object. In order to evaluate the robustness of our method with respect noise

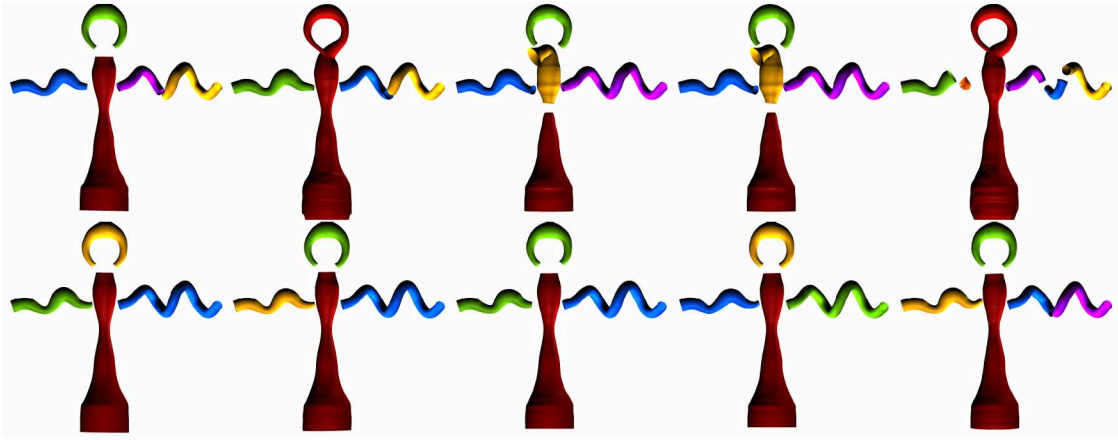
and undersampling, we generated several data sets: first we decimated the point clouds, generating new data containing only 75%, 50%, 25% and 10% of the initial clouds (Figure IV.9), and second, we added an amount of surface noise varying from 1 cm to 4 cm (i.e. ranging from 1% to 5% in thick parts of the model, and from 3% to 40% in thin parts) (Figure IV.10).

The tuboid reconstruction procedure took in average 110 seconds, including 1 second for computing the HT (0.5 second in the case of single-sided acquisition). Actually, many HS local maxima of low score used as seeds for tuboid growth are further discarded (the generated snake are non significant, correspond to to noise or conflict an existing tuboid). Therefore, fixing a maximum number of tuboids to be extracted avoids the analysis of such maxima and reduces the computing time to 50 seconds. With these results, the STEP algorithm proves its robustness for both scanning configuration (single-sided or full shape) even in the presence of noise or undersampling.

The results obtained from this data set provide material to analyse the ability of our algorithm to face complex models. They also illustrate the effects of the acquisition protocol (number of points of view), as well as those of the sampling resolution and homogeneity. Figure IV.10 confirms, that the surface noise impacts the reconstructions quality. Due to perturbations, the lack of local consistency in the computed normal vector field disturbs normals convergence. This either creates cuts in the reconstructed shapes (Figure IV.10d, top and IV.10e, top and bottom) or perturbs the stopping criterion, leading to a single snake covering multiple parts of the object (Figure IV.10b, top, and IV.10e, top). However, these results also underline the resilience of our approach ; even with the highest level of noise, the shape is globally properly reconstructed. The amount of missing points in the data also deteriorates the extracted tuboids. However, our tests present a significant quality of reconstruction up to 75% of missing points with respect to a common TLS acquisition sampling resolution. Multiple points of view enhance the reconstructions and its resilience compared to a single scan acquisition: reconstruction is stable in registered clouds up to 4 cm noise.

Third data set

The third data set was selected to apply the STEP algorithm to a natural forest point cloud. A tree was isolated by an operator from the original TLS point cloud by keeping the points of the stem along with 15 cm of the surrounding branches (Figure IV.11). Similar to the first test, the resolution of the corresponding HS was of 2 cm for the x, y, and z dimensions and 1 cm for the radius. After the *in situ* acquisition, the target tree was cut down, the branches were pruned, and the remaining log was scanned by a very high precision scanner. This procedure gave access to a reliable reference measurement of the stem taper. Comparison



(a) no noise (b) 1 cm noise (c) 2 cm noise (d) 3 cm noise (e) 4 cm noise

Figure IV.10. – Reconstruction of a tubular object scanned with increasing noise. The top row illustrates the results for a single scan while the bottom row shows the reconstruction from two point clouds acquired from different points of view.

between the STEP reconstruction and reference values provided a quantitative analysis on the error of the estimated stem taper (Figure IV.12). It is important to note that the reference diameter measurement did not include the tree bark. Hence the presented taper error incorporates a 0.4 cm bias.

This test showed how the STEP algorithm is adapted to TLS scans in natural forest. The measured error in the stem taper was mainly under 1 cm up to 6 metres from the ground. Beyond that point, the branches generated large occlusions and noise points (points located between two surfaces hit by a TLS laser beam). However, even above 6 m, the attractors inside the growing search cone were still sufficient to capture a general but less accurate description of the tree. The obtained results are in the range of acceptable forest measurements, which demonstrates the relevance of the algorithm on forestry data.

Fourth data set

Some TLS acquisitions of trees in a forest plot were taken from the SimpleTree© open data page [71] to demonstrate the capacity of the STEP algorithm to extract the main branching structure of the trees (Figure IV.13 and IV.14). Even though each tree was isolated in a single point cloud and automatically denoised, the registration of several acquisitions generated defects in the point clouds. The radius resolution of the HS was set to 1 cm and the x, y, and z resolution of the HS were set to 4 cm to decrease the running time compared to 2 cm. The main structures of 12 *Erythrophleum fordii* and 12 *Quercus petraea* trees were re-

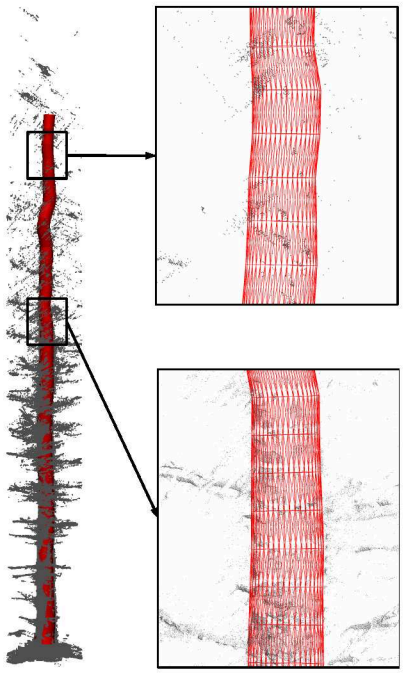


Figure IV.11. – Reconstruction of a noisy and occluded coniferous tree scanned with TLS. The resulting tuboid overcome these limitations to reconstruct the upper part of the stem.

constructed. The STEP method extracted the tree's structure within the range of 1min20s to 2min, including 2 to 5 seconds for the HS computation. Limiting the number of tuboids to be extracted to 10 reduced the running time between 30 and 50 seconds. As an additional robustness test, we ran the STEP method on the same data sets after randomly selecting only 50, 25 and 10% of the point clouds (Figure IV.13 and IV.14). The standard parametrisation of the method was not appropriate for only 10% of the points. We then modified the x, y, z resolutions of the HS to 5 cm in order to better observe the normals convergence.

Finally, these last tests established the ability of the STEP algorithm to retrieve the main structure of woody parts of tree in spite of TLS artefacts and subsampling. This way we indicate the potential of the STEP algorithm to measure stem taper, but also to furnish higher level information of interest in forest monitoring.

6. Discussion

We propose an novel algorithm using pattern recognition tools which reconstructs tuboids as single entities in order to obtain greater control on the output

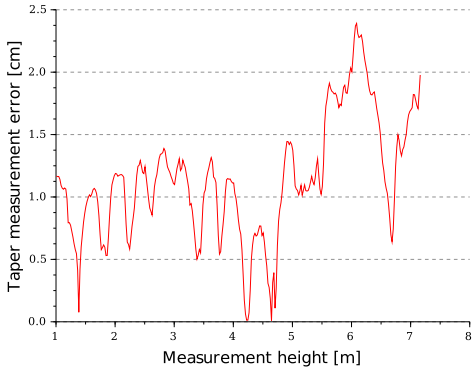
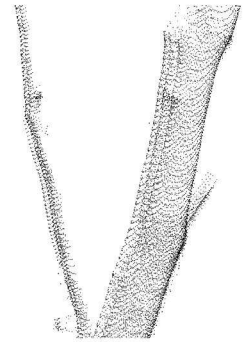


Figure IV.12. – Tapering error of the scanned tree reconstructed in Figure IV.11.

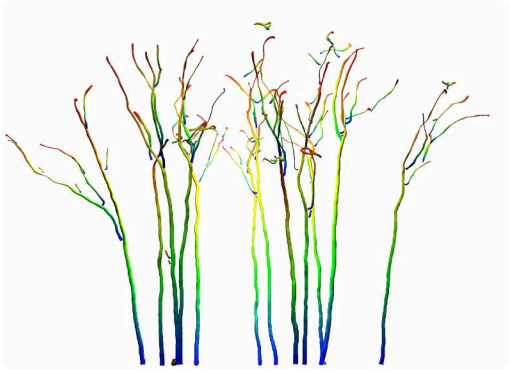
shape. We extended and combined two well-known mathematical tools that had not been exploited to their full potential to treat the tubular shape extraction issue. We first defined an original cylinder Hough transform to efficiently extract tuboids from a point cloud with normals. This innovation was accomplished by reducing the HS dimensions to 4 and developing a specific HS analysis, since the usual HS analyses do not match our objectives. We also generalized the active contours energy by taking into account curve parametrisation in the data energy term, and thus obtain an adapted open growing contour algorithm. This novelty has a large effect on the energy minimisation procedure. Results have illustrated the resilience of the STEP algorithm to limitations that are inherent to unstructured point clouds, and especially to TLS data, viz. different sampling densities within the point cloud, signal occlusion and presence of noise. Overall, average reconstruction errors were in the range of the used HS resolution. Our algorithm proves to be stable, to reconstruct complex shapes, and to be resilient to noise and shifts in sampling resolution with acceptable reconstruction until 75% subsampling. Moreover, the STEP algorithm is able to reconstruct cone shapes. In the context of tree reconstruction, the methodology is entirely different from what is currently used for stem taper estimation [74, 168, 180]. However, it shows robust competences to overcome the TLS limitations and presents an accuracy acceptable in operational forest inventories. Not only the main stem, but also the major structure can be reconstructed. Thus, we believe that with improvements, our algorithm has the potential to provide a complete QSM model of trees. Compared to other tubular reconstruction methods, the STEP algorithm offers additional advantages for dealing with noisy and occluded point clouds that are typical of data acquisition in natural forest environments (Figure IV.11).

For a better understanding of the STEP method, we provide more insights on its application. First, the STEP method aims at extracting tuboids from point clouds by observing normal convergence towards the centres of circles. However,

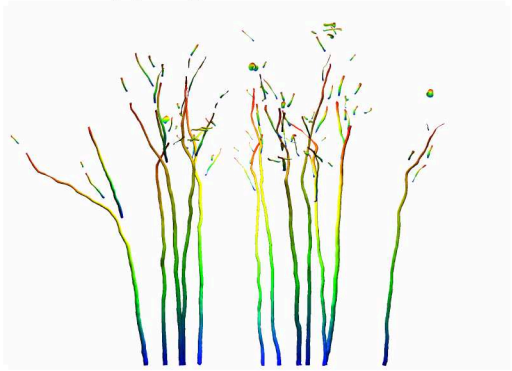
(a) Original cloud.



(b) Alignment artefacts.



(c) 100% of the points.



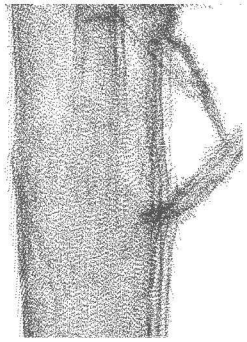
(d) 50% of the points.

(e) 25% of the points.

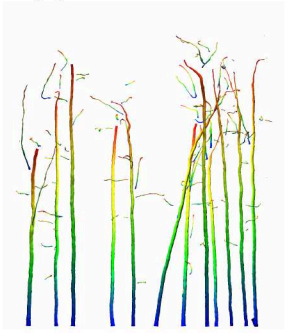
(f) 10% of the points.

Figure IV.13. – Reconstruction of the main branching structure of 12 *Erytrophelum fordii* trees with with increasing decimation of points. The wind combined to the point clouds registration produced misalignments.

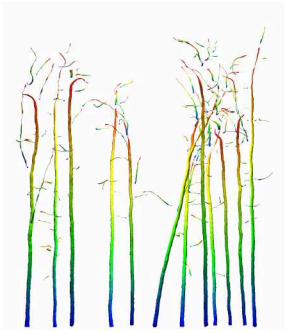
(a) Original cloud.



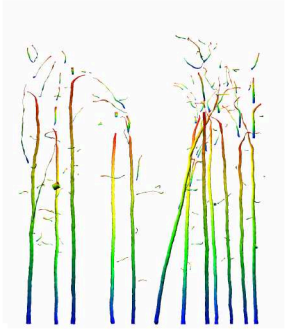
(b) Alignment artefacts.



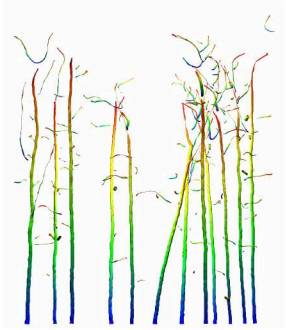
(c) 100% of the points.



(d) 50% of the points.



(e) 25% of the points.



(f) 10% of the points.

Figure IV.14. – Reconstruction of the main branching structure of 12 *Quercus petraea* trees with increasing decimation of points. The wind combined to the point clouds registration produced artefacts.

even though normal computation on point clouds has been previously studied, is still a challenging task. Especially with the geometry of forest structures which makes it difficult to estimate an accurate normal vector for each point. The quality of STEP method results suffers from inaccurate estimates of normals, since convergence of normals is not assured in such cases. Figure IV.11 actually shows that the STEP method's accuracy decreases when confronted to higher occlusions and noise in the upper part of the tree. This can be explained by the difficulty to obtain accurate normals under such conditions. However, when the point cloud is not affected by these limitations, the results have shown that optimisation-based normals estimation procedures constitute good candidates to overcome this issue. Also, like shape fitting procedures, the STEP method faces limitations when dealing with irregular (non-circular) tree stem shapes since normal convergence is not observable. Second, two conditions are required for consistent reconstructions with the STEP method: (1) HS resolution for the x, y and z parameters must be at least twice the HS resolution for the radius parameter, and (2) HS radius resolution must be below than the radius of the smallest radius to be reconstructed. These two conditions ensure that score accumulation due to the convergence of the normals will be observed. Very fine-scale resolution may not be adapted since the accumulation of scores in the HS can not be observed in such cases (because of errors on the normals or because of stem irregularities). The high dimensionality of the HS (4D) requires a large amount of memory, especially when the scene is large or the HS of high resolution. Thus, HS resolutions must be carefully adapted both to the input point cloud, as illustrated by the tests made on the fourth data set, and to computer performances. Third, the results analysis indicate that an *a priori* estimation about the number of tubular shapes to be reconstructed significantly limits the running time of the algorithm. Unfortunately, this information is not always known and interferes with the principle of a fully automated method. Fourth, the STEP method does not reconstruct the topology of the scanned objects. Instead, it generates numerous independent non intersecting tuboids. Regardless, through our tests at this point, the STEP method has demonstrated capabilities to extract tubular shapes and measure trees attributes from TLS data acquired in various sampling conditions.

The new developments that are presented here lead to several directions for future work. Indeed, a novel multi-scale approach could be set to improve efficiency of the STEP algorithm (both in running time and memory requirements). A more adapted growth stopping criterion could also increase the overall results. Finally, the next step for our algorithm is to handle the objects topology. The STEP algorithm can be used effectively even without these improvements. Hence, we plan to test it in operational contexts of forest inventory on diversified forest stand structures and finely assess its limitations before implementing further improvements.

7. Conclusion

We have proposed the STEP method as a new mathematical approach for reconstructing general tubular shapes with special attention to tree reconstructions from TLS data. We successfully extracted complex shapes from data affected by different levels of noise and subsampling, as well as tree stem taper from TLS data that were collected from forest plots. This method was developed to overcome the limitations of existing approaches, and especially for TLS acquisition in forest environment that hinder these approaches by their specificities (such as non-homogeneous sampling densities, signal occlusion and noise which are major issues in natural forest environments). We propose an original HT taking advantage of point normals to accelerate the computation and we defined generalised growing open active contours to incorporate curve parametrisation in the data energy. We thus introduce a clear and robust mathematical framework for the STEP method that allows automatic complex tubular shapes reconstruction and in particular, stem taper estimation. Tubular shapes are recovered as a curve in the 4D Hough space, which enhances the coherence of the reconstruction process. Another major advantage of the STEP method is its ability to handle the complete point cloud of a scene (and especially a forest scene) without the need to isolate objects for their reconstruction. The method can be applied to any unstructured point cloud and proved to be efficient for its favourite target, namely raw TLS point clouds, which are potentially composed of several aligned scans.

The STEP method opens several promising perspectives for future developments. The mathematical framework that we have adopted and developed is flexible, and therefore can be optimised for forest inventory or for other applications like the extraction of specific objects in urban or industrial scenes that have been acquired with TLS or photogrametry. An additional development will be to handle the structure of scanned objects. The current implementation of the method includes a non intersecting rule for the extracted tuboids. We actually expect to be able to design an appropriate method to connect and fuse some of the isolated tuboids into higher level objects and higher level representations. Furthermore, a multi-scale version of the STEP method is being designed to reduce the requirements of memory and computing time. Further tests are planned as the next steps towards validation within operational forest inventory. Finally, other tests should be performed within industrial and urban monitoring or refinery reconstruction.

V. Validation of the STEP algorithm

Overview

This chapter focuses on the extensive validation of the previously presented STEP algorithm and on its comparison with state-of-the-art algorithms. TLS forest data are well suited for the robustness evaluation of our proposed algorithm since they are strongly affected by noise and occlusions. In addition, modelling tree stems as tubular shapes is the standard practice during TLS-aided forest inventories (see Chapter I). The application of the STEP algorithm to forest point clouds is hence ideal for quantifying the algorithm capabilities to overcome the limitations of noise and occlusion while retrieving the geometry of scanned surfaces. Therefore we solve the lack of strong validation stated in the previous chapter and demonstrates the advantages of our algorithm in an applicative context with challenging data sets.

In this chapter, we first introduce the context of forest monitoring and we present the existing approaches for tree stem modelling as tubular shapes. Then, we provide some insights about two state-of-the-art algorithms developed specifically for the applicative context of forest point cloud processing. We also illustrate the strengths and weaknesses of each algorithm based on statistical results.

1. Introduction

Long-term preservation of forest resources can only be achieved based on accurate forest monitoring and with the help of precise knowledge. Implementation of sustainable practices for the management of forest resources implies the availability of reliable forest inventories. Precise measurements of forest attributes also contribute greatly to the understanding of complex ecological processes. Forest inventory is based on our ability to measure or estimate key tree attributes. Among these attributes the diameter at breast height (DBH, diameter of the trunk 1.30 m above ground) and the stem tapper (diameter of the trunk as a function of height) are of particular interest.

A reliable measurement of tree DBH within a forest plot is easily achieved by field crews. During traditional forest inventories, it can be directly measured by operators using calipers or diameter tapes. However, measurement made with such instruments include potential sources of errors [10, 100, 103]. Manipulation of the measuring instrument is subjective and operator's read on the gradation lead to potential error. For example, a angle of deviation of 2.5 degrees during diameter measurement using caliper generates an error estimated at 2 cm [103]. Moreover, the transcription of data from the instruments to the note book, and from the note book to a spreadsheet may introduce additional sources of error. Regardless of these potential errors, the measurement values are usually within operational margins for use in the forest inventory. The DBH measurement has become an essential part of the forester practice, it is now a reference variable for the development of allometric models. These models tie DBH with many useful structural variables, namely tree height [61], merchantable or total wood volume [66], biomass [205], or total leaf area [193].

With contrast to the DBH, stem tapper is more difficult to determine in the field as field operators can hardly measure it in situ in a non destructive manner, because of logistical issues. For instance, measuring a standing tree requires that a field operator climbs on the trees, therefore limiting the height at which diameter measurements can be taken on the stem. In addition no optical instruments operated from the ground can currently provide a reliable measure of stem taper and overcome this laborious procedure. Consequently, stem tapper can not be consistently measured during forest inventory campaigns with non-invasive methods. Different solutions may be considered to acquire this attribute: it can be either measured manually through a destructive protocol or estimated with the help of precise scanning technologies.

Destructive protocol to measure stem taper involves cutting the trees of interest into logs. Circumference measurements can then be taken on the logs to

obtain the stem taper. Specific field campaigns were designed to include tree cutting to measure tree stems. Cutting tree to measure stem taper is not an acceptable sustainable practice for most forest inventory plots because of ecological and logistical issues. For instance, it is incompatible with remeasurements of permanent plots. Hence destructive sampling can only be considered exceptionally for specific purposes. Consequently, destructive measurements of stem taper measurement can not be considered as a practical solution for regular forestry inventory plots. Considering the importance of stem taper to estimate the useful portion of the stem for wood supply, an efficient solution remains to be developed for the estimation of stem taper without requiring destructive sampling.

Laser scanning technologies are gradually becoming an essential non-invasive solution to monitor forest environments [47]. LiDAR (Light Detection And Ranging) instruments provide 3D point clouds of their environment containing several tens of millions of points. Two major types of LiDAR instruments have been used to monitor forest environments [200]. Airborne laser scanners (ALS) are mounted on an airplane flying over the study area. While it is possible to derive some structural attributes from ALS data, estimation methods are mainly designed for stand-level measurements (e.g. aboveground biomass [31]) and for tree attributes readily available from above (e.g. tree height [59] ; crown area [144]). Tree stems are poorly probed with ALS systems because of signal occlusion by the upper canopy. In addition the ALS point cloud is generally not dense enough to retrieve stems taper accurately. In contrast, terrestrial laser scanners (TLS), which are located on the ground, probe their local environment using a hemispherical sampling pattern with up to several hundreds of millions of returns, resulting in high quality point clouds. TLS has the ability to capture accurately the shape of the stems up to the upper part of the tree, depending on the amount of occlusion created by the lower branches of the tree. However, TLS data include some limitations such as signal occlusion, noise (signal returns not related with a scene object) and non homogeneous sampling density. Specialized point cloud processing algorithms are therefore required to estimate tree attributes from TLS point clouds [114]. Such automatic algorithms are objective and repeatable. Also, these algorithms lead to systematic errors instead of subjective errors related to manual measurements. The use of TLS point clouds for the estimation of tree attributes is attractive for several reasons. Firstly, it does not require destructive sampling. Secondly, it allows making estimations with an acceptable range of error: the accuracy reached by the TLS data-driven algorithms has the potential to be comparable with high resolution scanning and software used in the wood industry. Thirdly, TLS data can also be used to estimate several tree attributes that can not be measured on field, such as crowns shape and competition [128, 133], plant area index [13, 55, 83], tree architecture [44, 45] or wood fibre attributes [24]. The potential for the use of TLS in forestry hence lar-

gely depends on the ability of algorithms to process efficiently the point clouds and to deal with the limitations of TLS data.

Several algorithms have been developed to estimate tree structural attributes from TLS point clouds [114]. Stem DBH and taper can be estimated from purpose-specific algorithms. Most of these algorithms assume that a stem cross section is of circular shape. Thus they aim at reconstructing stems as a series of circles or pieces of surfaces of revolution (namely cylinders and cones). Circle fitting has been applied successfully on point clouds to retrieve stem DBH and taper. These approaches use various clustering methods to select appropriate points in layers before applying either least square circle fitting [22, 33, 78, 102, 118, 152, 180], RANSAC algorithm [151], or other methods [98] to retrieve the stem centre position and its radius. In particular, the Computree software platform developed by the ONF provides an advanced applicative algorithm to estimate tree DBH and taper based on least square fitting [152]. These studies show radius estimation error in the range of 1 cm to 3 cm, however they tend to provide erroneous models when the point cloud suffers from occlusion and noise. The more general ellipse fitting has also been tested [1, 8, 15, 154]. However the ellipse fitting tends to be sensitive and unstable [2]. Cylinders have been employed to tackle the issue of circles orientation and the choice of appropriate layers [142, 203, 209]. Iterative least square cylinder fitting have been set up [161] with robust variants using normal information [115, 116] or MSAC (M-estimator SAmple and Consensus) [94]. Despite the accurate results, cylinder-based methods appears to suffer from the same issues as circle-based methods. Tree stem reconstruction as a series of cones has been introduced to estimate the local tapering of the trunk [95, 131] with no major effect on the robustness of the methods. Other approaches have been employed either for retrieving a more precise perimeter estimation via b-spline fitting [161], or for approximating tree volume by using polygons [79] that could easily be adapted to estimate stem perimeter. These method result in precise estimation but requires all the stem to be probed, and hence cannot be applied on single scan data. Another approach to estimate DBH from TLS data implies using the Hough transform (HT) to identify circles along the stem [186]. As a powerful pattern recognition tool, the HT shows robustness to both occlusion and noise. Since then, it has been successfully used as a pre-localisation tool or in combination with classical circle fitting [9, 85, 118, 151, 180]. Many solutions for tree DBH and taper estimation have been proposed, nonetheless as stated in [206], occlusion and noise affect most of the presented algorithms and robust methods are still in demand.

In an attempt to overcome the TLS limitations affecting most algorithms to estimate stem taper, we developed a new algorithm based on a HT variant called STEP (Snake for Tuboid Extraction from Point cloud) [171] which is described in detail in the previous chapter. The STEP algorithm was designed to estimate tree

DBH and taper in natural forest environments. Stem taper can also be estimated by using TLS point clouds with algorithms designed to simulate tree architecture [44–46] or with the quantitative structural models (QSM) [72, 74, 168, 169]. These QSM, including the SimpleTree software [73], use cylinders to provide a complete description of a tree, including architecture, stem and branches diameters and total volume. However the accuracy of the description depends upon the quality of the input point cloud. Allometric relationships are used to correct erroneous estimations due to the poorly described geometry of the stems [73]. There is a need to benchmark these methods in order to better understand their respective advantages and to evaluate their accuracy in various natural forest environments.

This study evaluates the capabilities of the STEP algorithm on a diversified set of TLS acquisitions in natural forest environments, and to benchmark the results with other algorithms designed to estimate DBH and stem taper. Consequently, the main objective of this study is to assess the accuracy of the STEP algorithm when applied to a range of TLS acquisition protocols for stem taper estimation. This objective was reached through two specific objectives. Firstly, we aimed at analysing the results of the STEP algorithm on several contrasted sites representing natural forest environments. Each site was related to a reliable but different set of reference measurements of stem taper which allowed us to compare the estimated values with well documented measurements. Secondly, aside from using the reference measurements on stem taper, we compared the results of the STEP algorithm with two other methods for estimating DBH and stem taper. Therefore we compared the reference measurements of stem taper with three specific algorithms using TLS data: (1) the STEP algorithm [171], (2) the advanced cylinder fitting algorithm available in the CompuTree software platform, and (3) the SimpleTree QSM algorithm [73].

2. Benchmarked algorithms for stem taper estimation

The three algorithms selected to estimate DBH and stem taper for the benchmarking represent three distinct strategies to deal with this problem using TLS data. These three algorithms are all implemented as plug-ins within the CompuTree software platform. CompuTree is a software platform created by the R&D department of the French National Forest Office (ONF). Its goal is to regroup TLS point cloud processing algorithms for forest scenes analyses. It provides a full development environment based on powerful c++ libraries, and as a collaborative platform it allows to implement various methods as plug-ins for comparative

and exchange purposes. Since, the STEP algorithm was previously detailed, we now provide provide some insights about the two other algorithms selected for benchmarking.

2.1. A cylinder fitting algorithm

The CompuTree software platform includes a DBH and stem taper estimation algorithm developed by the ONF based on the work of [152]. It will be referred to as the CompuTree cylinder fitting algorithm (or CCF algorithm) from this point on. The CCF algorithm was selected because it represents the family of algorithms that are using cylinder or cone fitting to estimate stem taper [114, 131, 161].

The CCF algorithm starts by applying a convolution filter to the original TLS point cloud to remove the branches of the tree and allow focusing on the stem for the remaining of the procedures. Then the algorithm divides the point cloud into horizontal layers of fixed thickness. An Euclidean clustering segments each stem individually in these horizontal layers. A cylinder is then fitted to the segments by retrieving the orientation of the cluster and using a least square optimization. More details about this stem DBH and taper estimation can be found on the Computree platform website [48].

The CCF algorithm was designed to require little computational time when applied to a forest plot while being able to provide accurate estimations of trees locations, tree DBH and stem taper. It has already proven its efficiency with high rates of stem detection [152]. Even though its parameters are intuitive, a calibration of the CCF algorithm is necessary to adapt it to the point cloud specifications and the forest stands characteristics. For instance these parameters include different thresholds applicable to the convolutional filter, to the euclidean clustering and to the least square fitting. The clustering step is sensitive to noise. In addition the shape fitting faces limitations where the stem splits into two or more branches [172]. Even though the CCF algorithm can not detect horizontal branches at this time, it is adapted to be use with TLS point cloud to estimate stem DBH and taper in forest inventory applications.

2.2. SimpleTree

We selected SimpleTree, a QSM algorithm developed by Hackenberg [73] as a third algorithm for our comparison study since it is a state-of-the-art QSM algo-

Study site	Newfoundland	Phalsbourg	Bas Saint-Laurent
Stand type	Coniferous	Mixed (mainly deciduous)	Coniferous
Tree species	Black Spruce Balsam Fir	Common Beech Hornbeam	White Spruce
Number of selected trees	24	40	17
Minimum DBH [cm]	5.43	10	9
Maximum DBH [cm]	19.88	70	40

Table V.1. – Overview of the data sets characteristics for the three test sites.

rithm. The version used was available as a plug-in on the CompuTree platform. Hence, the all the three algorithms selected were implemented in the CompuTree platform. It facilitated a rigorous and homogeneous evaluation based on the same operating platform. A short summary of the SimpleTree method follows.

The SimpleTree algorithm provides a geometrical structural model of a tree, including branches, formed by interconnected cylinders generated from circle fittings. It starts modelling the tree from the estimated circle C0 at the base of the stem. The centre of C_0 is used to position a cutting sphere S_0 with a higher radius than C_0. The intersection between S_0 and the input point cloud represents a cross-section of the stem. A circle C_1 is then fitted to the cross-section points and a new sphere S_1 is generated at the centre of C_1. The stem is iteratively reconstructed by moving the cutting spheres S_i along the fitted circles C_i. When a cutting sphere intersects several cross-sections, a new branch has been detected and will further be reconstructed with the same procedure. The results of the SimpleTree algorithm are available as hierarchical series of cylinders derived from the fitted circles C_i.

The SimpleTree algorithm runs iteratively until it retrieves the entire tree structure including branches diameter and order. It has been enhanced with automatic parametrization, optimization and forestry knowledge such as allometry. However, it requires the tree to be isolated in a single point cloud. Even if it has been validated with several TLS data sets, it may be useful to use SimpleTree in a benchmarking exercise to develop an extended understanding of its advantages and limitations in a wide range of natural forest stand acquisitions.

(a) Isolated tree (b) Zoom on branches (c) Isolated tree (d) Zoom on branches

Figure V.1. – Examples of simulated point cloud from the Newfoundland data set. Even though the point clouds were noise free, they suffered from branches occlusion.

3. Test sites and data sets

Since stem tapper is rarely measured because of the required destructive samplings, only few forest sites matched our ground truth requirements for benchmarking. Nonetheless, we selected three test sites with different, yet accurate, reference measurements to evaluate the three selected algorithms for DBH and stem taper estimations. The sites were selected to provide diverse structural configurations and different data quality to meet our study objectives. The three sites offer a variety of locations, forest environments, species, data sources, and acquisition protocols. Reference measurements were available for each site but came from different sources. The three algorithms (STEP, CCF algorithm and SimpleTree) were applied on a tree-level to 81 isolated trees: 24 from the Simulated forests data set replicating forest plots in the Newfoundland and Labrador province (Canada), 40 from the Phalsbourg data set collected in the Lorraine Département (France), and 17 from the Bas Saint-Laurent data set collected in the province of Quebec (Canada). The three test sites provided reference measurements of stem taper adapted to validate the three algorithms on a large number of trees in simulated and natural environments. They cover both coniferous and deciduous trees and include multiple species. These data sets also provided various limitations with leaf-on scans and various degrees of occlusion and noise. Table V.1 presents a summary of the trees selected for evaluation on each data set.

3.1. Simulated forest stands from plots in Newfoundland

The first data set is a collection of three simulated forest plots that were populated with tree models obtained from the architectural model L-Architect [44–46]. This model uses TLS scans of in situ trees to build a catalogue of 100 trees or more of a given species to represent typical architectural traits. The catalogue can then be used to simulate forest stands. Tree selection in the catalogue was based on three attributes, namely their tree DBH, height and total leaf area. Trees of plots scanned by TLS are first isolated and then tree that fits the most from the catalogue is selected and rescaled to fit the space of the in situ tree. L-Architect was used for an accurate simulation of forest stands containing two tree species: black spruce (Picea mariana (Mill.) BSP) and balsam fir (Abies balsamea (L.) Mill.). Three virtual forest plots with an area of 900 m2 (square plots with 30 m edges) were generated and were comprised of 30, 71 and 92 trees respectively. We selected randomly 24 virtual trees with DBH ranging from 5.43 to 19.88 cm in the three simulated plots for the benchmarking (Figure VI.4).

TLS acquisitions of these virtual forest plots were simulated using a physically based ray tracing tool [162] using the following typical scan settings: (1) all scans used an angular resolution of 0.036°for rays send throughout the full hemisphere, (2) rays are dimensionless vector with no divergence, and (3) the full scene point clouds were made of 5 scans: one in the centre of the plot, and 4 positioned at each extremity of the plot (north, south, east, west). The ray tracing simulations resulted in noiseless point clouds in which the soil was absent since it was not modelled in the virtual plots provided by L-Architect. However, the trees included in the virtual plots had a large number of branches. This generated numerous occlusions at various degrees on the tree stems. The trees were later isolated in the resulting point clouds in order to apply the stem tapper estimation algorithms at tree-level.

The simulated TLS acquisitions of virtual forest plots are an approximation of real scans. However, the use of modelled trees provides absolute reference values for tree attributes. Indeed the tree models provided the exact information on DBH and stem tapper. Absolute reference stem taper was available every 10 cm starting from the tree base. Since the point clouds were noiseless and the modelled stems perfectly cylindrical, the occlusion was the main limitation for the algorithms to overcome. Hence, these simulated plots were designed to evaluate the advantages and limitations of the stem taper estimations in an ideal but occluded environment.

(a) Isolated tree (b) Zoom on branches (c) Isolated tree (d) Zoom on branches

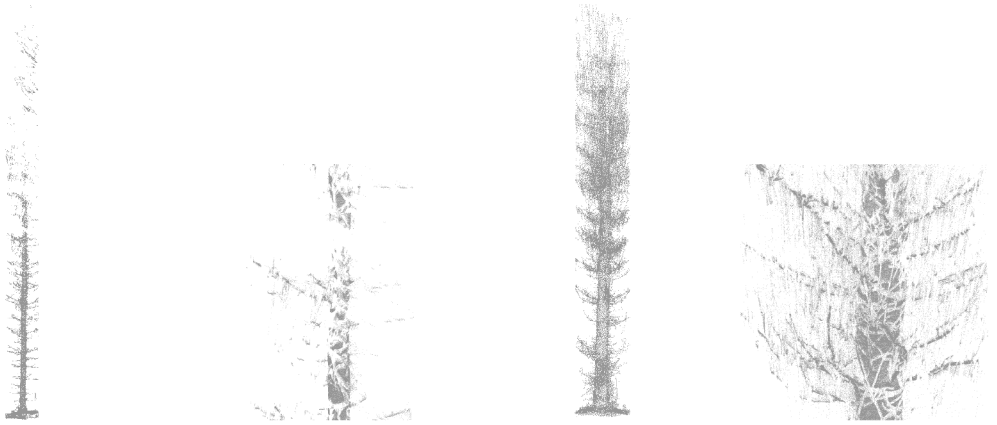
Figure V.2. – Examples of point cloud acquired at one of the Phalsbourg test sites. The point clouds exhibit large occlusions on the stem.

3.2. Phalsbourg data set

The second data set contains several TLS point clouds acquired in natural forest environments. It was collected during a field inventory campaign in 2013 by the R&D department of the ONF. The study sites were located near the city of Phalsbourg in France. The permanent forest plots (circular plots with a 15 m radius) selected included 12 different stands, all dominated by deciduous tree species, with a predominance (in decreasing order) of common beeches (Fagus sylvatica) and hornbeams (Carpinus), and in fewer number there were Norway spruce (Picea abies (L.) H. Karst), Scots pine (Pinus sylvestris L.) and silver firs (Abies alba Mill.). The plots were scanned during leaf-on season and the DBH of inventoried trees ranged from 10 to 70 cm. We selected randomly 40 of these trees with DBH from 15 to 60 cm to validate the three selected algorithms (Figure V.2).

The TLS scans were acquired with a Faro Focus 3D©scanner with an angular resolution of 0.036°. Each plot was scanned from a central position as well as three other locations in an equilateral triangle 7.5 m away from the centre of the plot. The scan settings induced a non homogeneous sampling density, and the presence of foliage during the acquisitions resulted in numerous occluded areas. The presence of noise was located in the upper part of the trees. The selected trees were then isolated into individual point clouds by using a cylindrical extraction of points centred at their corresponding coordinate on a tree map.

During the field campaign, inventoried trees were cut off at breast height.



(a) Isolated tree (b) Zoom on branches (c) Isolated tree (d) Zoom on branches

Figure V.3. – Examples of point cloud acquired with a terrestrial lidar at the Bas Saint-Laurent test sites composed of white spruces. The point clouds exhibit significant noise and occlusions on the stem.

Stems were then fell and further cut into 2 m long logs. The diameter at the end of each log was measured manually twice with a caliper (measurements taken at right angles). Thus, starting from 1.30 m above ground, manual measurements were available at each 2 m from the tree base to served as reference values for stem taper. With such data, the results of the three methods under evaluation on natural forest sites could be compared to traditional field measurements. Moreover, running the three algorithms on this data set evaluated their ability to overcome TLS limitations such as occlusion, noise and non homogeneous sampling density in natural forest conditions.

3.3. Bas Saint-Laurent data set

The third data set groups several TLS point clouds acquired in a plantation. During the fall 2014, a research team from the Université de Québec à Rimouski led the field campaign in the natural reserve of Matane (Quebec). The plantation of white spruces (Picea glauca (Moench) Voss) was located in the balsam fir-yellow birch bioclimatic domain [165]. A plot of 60 m $\times$ 15 m was set up in the plantation and divided into four 15 m $\times$ 15 m subplots. The resulting subplots included trees with DBH ranging from 9 to 40 cm. We selected randomly 17 trees with DBH from 9 to 37 cm to be processed for the evaluation of the three stem taper estimation algorithms.

The site was scanned using a Faro Focus 3D A120© scanner with an angular

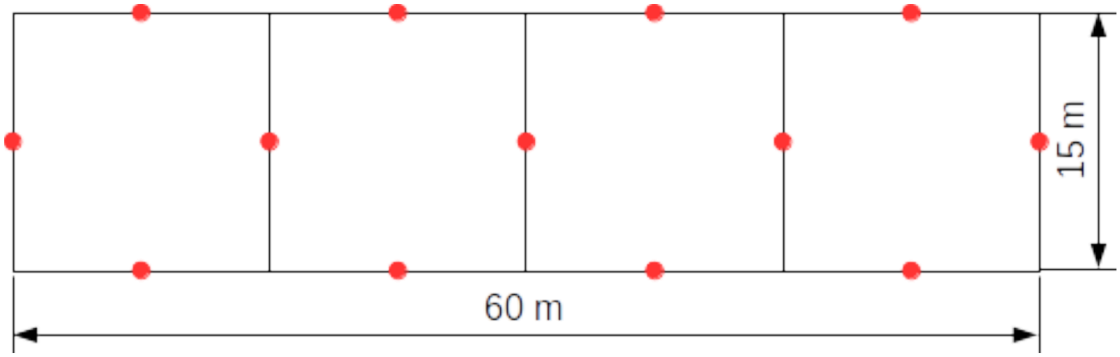


Figure V.4. – Scanning set up adopted for the terrestrial lidar acquisition at the Bas Saint-Laurent test site. Each of the four sub-plots (black) have been scanned from 4 different positions (red).

resolution of 0.018°. For each subplot, 4 scans were acquired at the plot borders (7.5 m away from the subplot centre) resulting in a total of 13 scans (Figure V.4). The resulting point clouds included abundant occlusions and noise due to the dense vegetation (branches and needles). The selected trees were manually isolated into separated point clouds for further processing (Figure V.3).

A linear carriage scanning system, called the True-Shape© scanner, was used as an alternative to manual measurement of stem taper (and DBH) in order to obtain reference measurements. The True-Shape scanner was designed to be a transportable and was developed to scan cut logs without branches. Once the trees of interest are fell from their base, the stems are pruned from their branches and the bark is removed. The True-Shape unit produces a high definition 3D point cloud of the outside shape of tree stems providing an accurate representation of the stem, from the cut and towards the tip of the stem. We used this representation to compute the best fit circle at each recorded height in order to obtain accurate reference measurements of the stem taper. However, while stems bark were removed, we observed an average bias of 1 cm between the original bark external surface of the standing stem and the diameter of the scanned stem. This third data set allowed us to validate the three algorithms in real coniferous plantation sites and to analyse their resilience to occluded, and noisy point cloud data.

The three sites presented here provided sufficient material of a quality adapted to validate the three algorithms on a large number of trees in simulated environments as well as in real conditions. They cover both coniferous and deciduous trees and include multiple species, which reinforce the pertinence of the evaluation of DBH and stem taper algorithms on such data sets. Similarly, they provide various limitations with leaf-on scans and various degrees of occlusion and noise. Each data set contains a reliable and accurate reference measurement

which is relevant for the analysis of the estimation algorithms results.

4. Methods

We proceeded in four steps to compare the three selected algorithms. For the first step, we calibrated the algorithms on a subset of the selected trees. Hence the algorithms were compared in a general framework rather (independent on the study plot characteristics). For the second step, we ran the three calibrated algorithms on the all the trees selected randomly for analysis. When necessary we eliminated values estimated from branches instead of the stem as a post-processing analysis. For the third step, we transformed the set of discrete reference measurements and diameter estimations at various height into a continuous taper function using linear interpolation. Finally, for the fourth step, we chose an error metric to compare the results of each algorithm against the reference values.

For the first step of our methodology we calibrated the three algorithms to achieve accurate results for each study site. Each algorithm is available as a script including a chain of processes (we refer to the CompuTree platform website for more details about these scripts and processes). The chain of process of the STEP algorithm includes 2 procedures: the computation of the accumulator space, and the use of open growing active contours. The size of the growing search cone and the value of the stopping criterion were the two major parameters to fix in this algorithm. As for the CCF algorithm script includes a convolution filter, an Euclidean clustering and a least square cylinder fitting. The most important parameters of the CCF chain were those of the convolution filter, the threshold distance used during the Euclidean clustering, and the RMSE threshold for accepting or discarding a fit cylinder. Now for the Simpletree script, it applies several processes: a point cloud filter, a statistical outlier removal, a voxel grid filter, and the sphere following algorithm. The statistical outlier removal and the voxel grid filter are the main processes that needed to be calibrated since the intrinsic parameters of the SimpleTree algorithm are auto-calibrated. Once the main parameters to fix have been identified for the three algorithms, a small subset of 9 trees (3 from each study site) were randomly selected for calibrating the algorithms. We calibrated the parameters of each algorithm using these test trees: the default parameter values proposed in the three CompuTree plugins were selected as a first set of parameter. The results of this set of parameter were then visually analysed, which guided us to chose a new set of parameter values. This procedure was iteratively repeated until convergence of the visual results on the 9 test trees.

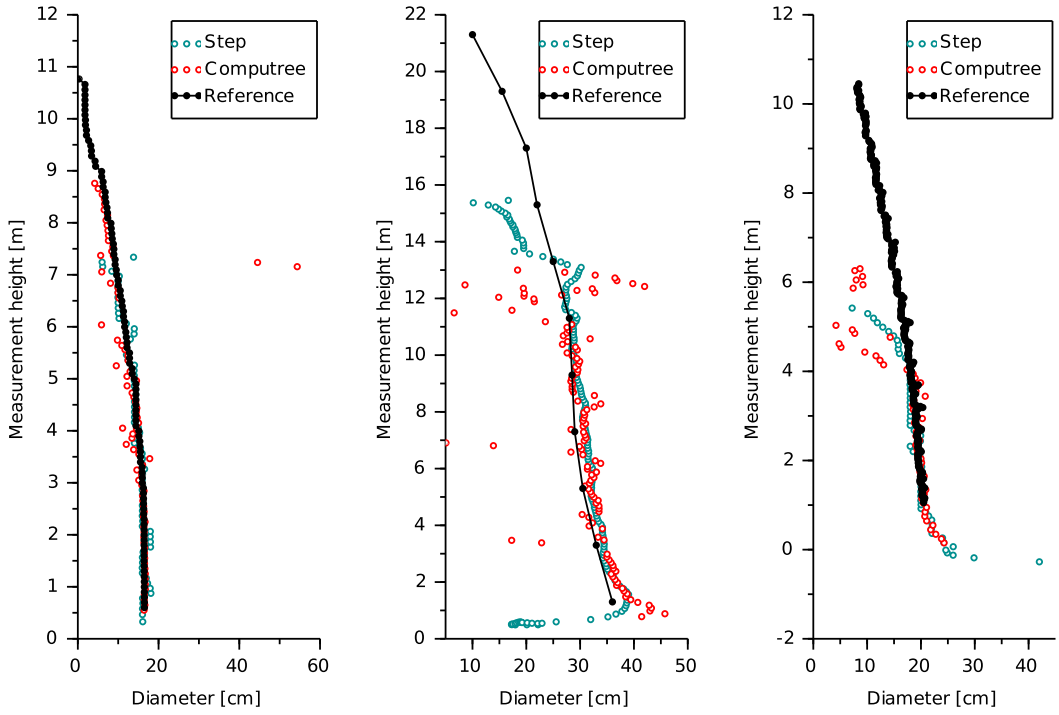


Figure V.5. – The reference taper functions are obtained using a linear interpolation on the reference measurements for a sampled tree of the Newfoundland (left), Phalsbourg (centre), and Bas Saint-Laurent (right) test sites.

As the second step of our methodology we applied the algorithms to all the study trees with the set of parameters adopted in the first step. However, the algorithms also provided estimation of diameters on branches in addition to the estimations on the stems. Since only the stem tapper estimation was required for the comparison of the algorithms with the reference measurement, we applied a post-processing analysis, based on visual assessment, to remove the values related to branches. In the case of the STEP algorithm, branches were avoided by limiting the active contour seeds in the range of 1 m to 2 m above ground. In the CCF algorithm, even though most of the tree foliage and branches were filtered by the convolutional filter, some points belonging to elements other than the stem remained and generated irrelevant cylinders. These cylinders were manually filtered. The Simpletree algorithm produces a complete reconstruction of the trees, including branches and tree structure. However, the branches hierarchy of the tree may contain errors: the labelling of the reconstructed cylinders may contains commission errors (cylinders wrongly identified as trunk) and omission

errors (stem cylinders identified as branches). Hence we inspected the output of the algorithm to select only the cylinders belonging to the stem for further evaluation.

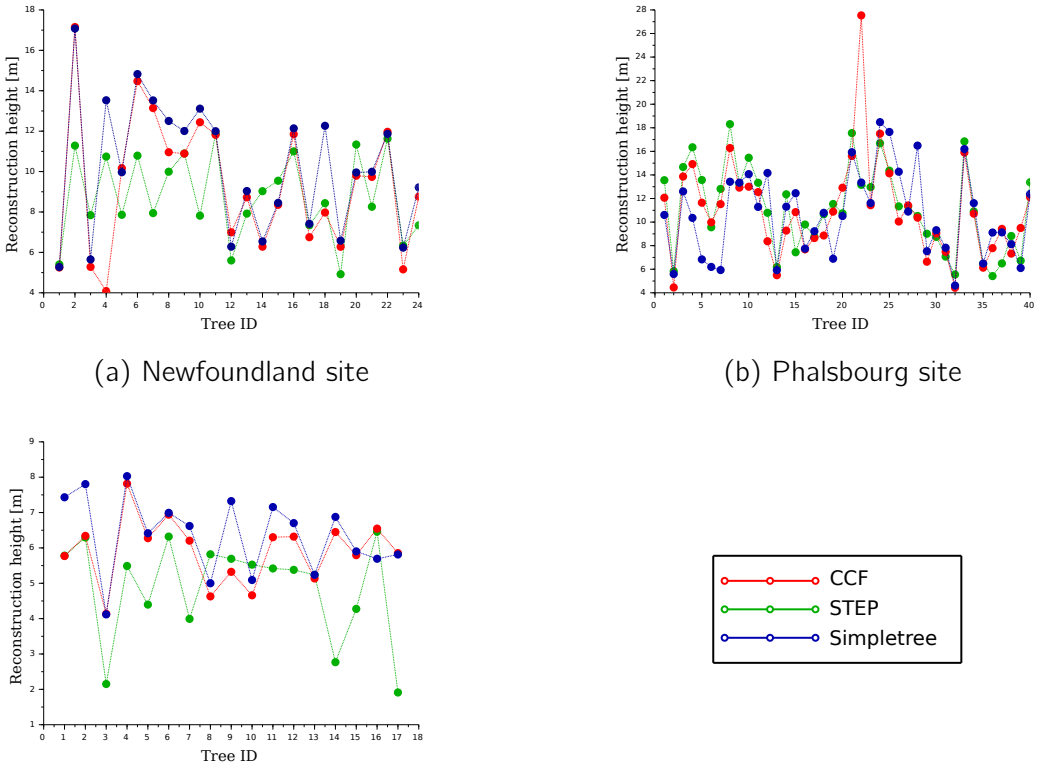
As the third step of our methodology we transformed the discrete diameter measurements into continuous taper function to facilitate results comparisons. For each tree, the continuous reference taper measurement were obtained with a piecewise linear interpolation of the reference diameters at each successive height. The height increment at which a reference diameter value was measured differed for each test site: 10 cm for the simulated forest plots, 2 m for the Phalsbourg data set and 1.6 cm for the Bas Saint-Laurent trees (Figure V.5). A similar linear interpolation was also applied to all estimated diameter values from the three algorithms. We preferred using linear interpolations instead of adopting allometric models. This choice was based on simplicity and avoided complex issues related with allometric modelling. However, adopting an allometric relationship is also another option not explored in this study. There are many models to pick from and in our study too few individuals were available to allow an accurate calibration of these models. Moreover, considering the available measurements height intervals,and that the curvature of taper models is usually low, the linear interpolation is still accurate since and it can be seen as a first order limited development. The stem diameter estimations, derived from the three selected algorithms, were transformed into a continuous taper function using linear interpolation between two consecutive circles (for the STEP algorithm), or cylinders (for the CCF and the Simpletree algorithms) (Figure V.5).

As the last step of our methodology we selected metrics to estimate the stem taper estimation error. To do so, both reference and estimated continuous taper functions were regularly resampled, every 2 cm of height. Then for each sampled height h_i, the error $E(h_i)$ was defined as the absolute difference between the reference diameter measurement $d_r(h_i)$ and the estimated diameter $d_e(h_i)$:

$$E(h_i) = |d_r(h_i) - d_e(h_i)| \tag{V.1}$$

5. Results

For each data set, we computed the stem DBH and tapering errors for the selected trees and evaluated the capabilities of the three selected algorithms. We computed the tapering error function for each tree, its minimum, maximum, mean and standard deviation. Besides these general statistics, we also considered



(a) Newfoundland site

(b) Phalsbourg site

(c) Bas Saint-Laurent site

Figure V.6. – Maximum height recorded for the estimation of stem taper of all trees in each test site and for the three algorithms.

the distribution of the tapering error (1) per tree and (2) per height intervals of 20 cm. In addition, we recorded the maximum height at which the algorithms could provide stem taper estimation for each tree.

The maximum measurement height for each simulated tree and each method is illustrated in Figure V.6 and more details are provided in Table V.2. The CCF and SimpleTree algorithm reconstructed the tree stems of the Newfoundland site up to the tree top. The tuboids produced by the STEP algorithm however reached lesser heights. On the Phalsbourg data set, the three algorithms reconstructed the stems up to similar heights. Finally, the STEP algorithm did not reach the same reconstruction height as the CCF and the SimpleTree algorithms on the trees from the Bas Saint-Laurent site.

The accuracy of the DBH estimation for each algorithm is illustrated in Figure V.7. The three selected algorithms achieved the same accuracy regarding the tree DBH estimation. However, the STEP algorithm tended to produce better the

Reconstruction height [m]	Algorithm	Newfoundland	Phalsbourg	Bas Saint-Laurent
Mean	CCF	9.35	11.02	5.91
	SimpleTree	10.22	10.66	6.37
	STEP	8.80	11.21	4.88
Std deviation	CCF	3.29	4.17	0.92
	SimpleTree	3.22	3.61	1.09
	STEP	2.09	3.58	1.42

Table V.2. – Reconstruction height of the three algorithms.

results than the CCF and the SimpleTree algorithms on the Phalsbourg and the Bas Saint-Laurent data set.

The measured tapering error are shown for each study site in Figure V.8, and Table V.3 summarizes the associated statistics. These statistics, along with the figure, show that the STEP method produces an average error per tree below the errors obtained with the other two algorithms, except on the Newfoundland site where SimpleTree generates more accurate results. The CCF algorithm however performs less accurately on every site of study.

Results also show that the error recorded by the three algorithms tends to increase with the measurement height when applied to real data sets. On the Phalsbourg and Bas Saint-Laurent sites, the STEP algorithm performed better than the CCF and SimpleTree algorithms in the lower part of the trees.

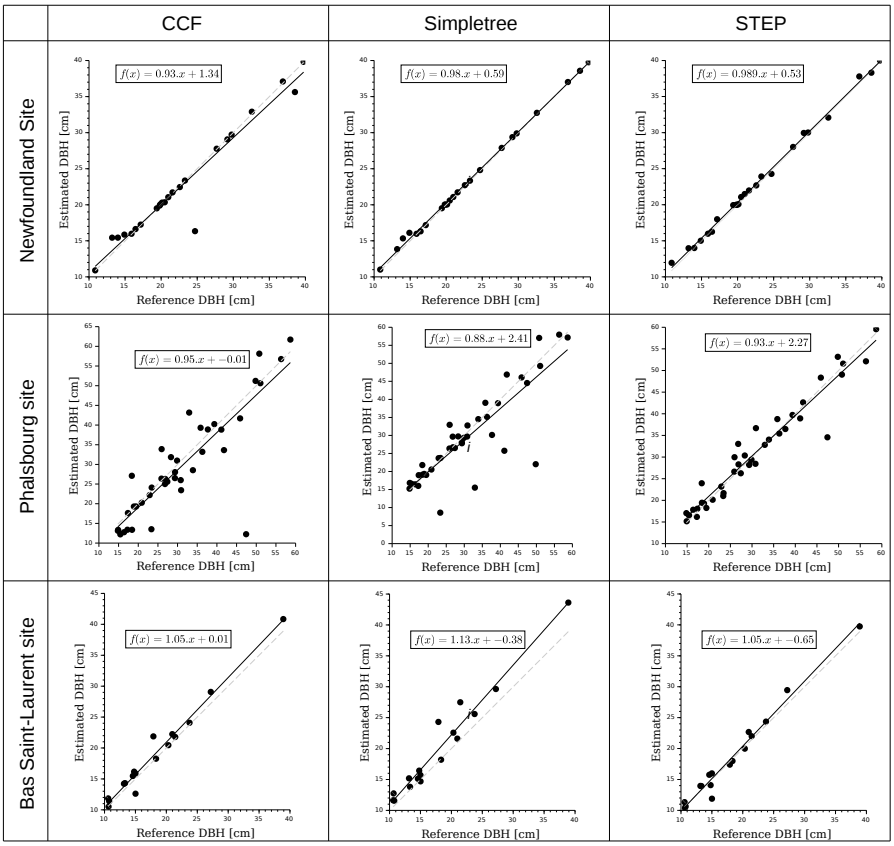


Figure V.7. – DBH estimation error using the three selected algorithms for each site. The grey dotted line is the 1:1 line. The linear regression equation is also given for the fit with the estimated values.

Taper error per tree [cm]	Algorithm	Newfoundland	Phalsbourg	Bas Saint-Laurent
Minimum	CCF	0.69	0.96	0.83
	SimpleTree	0.24	0.95	0.77
	STEP	0.37	0.93	0.47
Maximum	CCF	7.39	21.65	5.37
	SimpleTree	3.81	11.87	7.35
	STEP	4.88	5.93	3.57
Mean	CCF	2.62	6.1	2.64
	SimpleTree	1.03	3.34	2.67
	STEP	1.11	2.28	1.50
Std deviation	CCF	1.44	3.96	1.35
	SimpleTree	0.85	2.88	1.72
	STEP	1.01	1.25	0.87

Table V.3. – Results of taper estimation with the three selected algorithms.

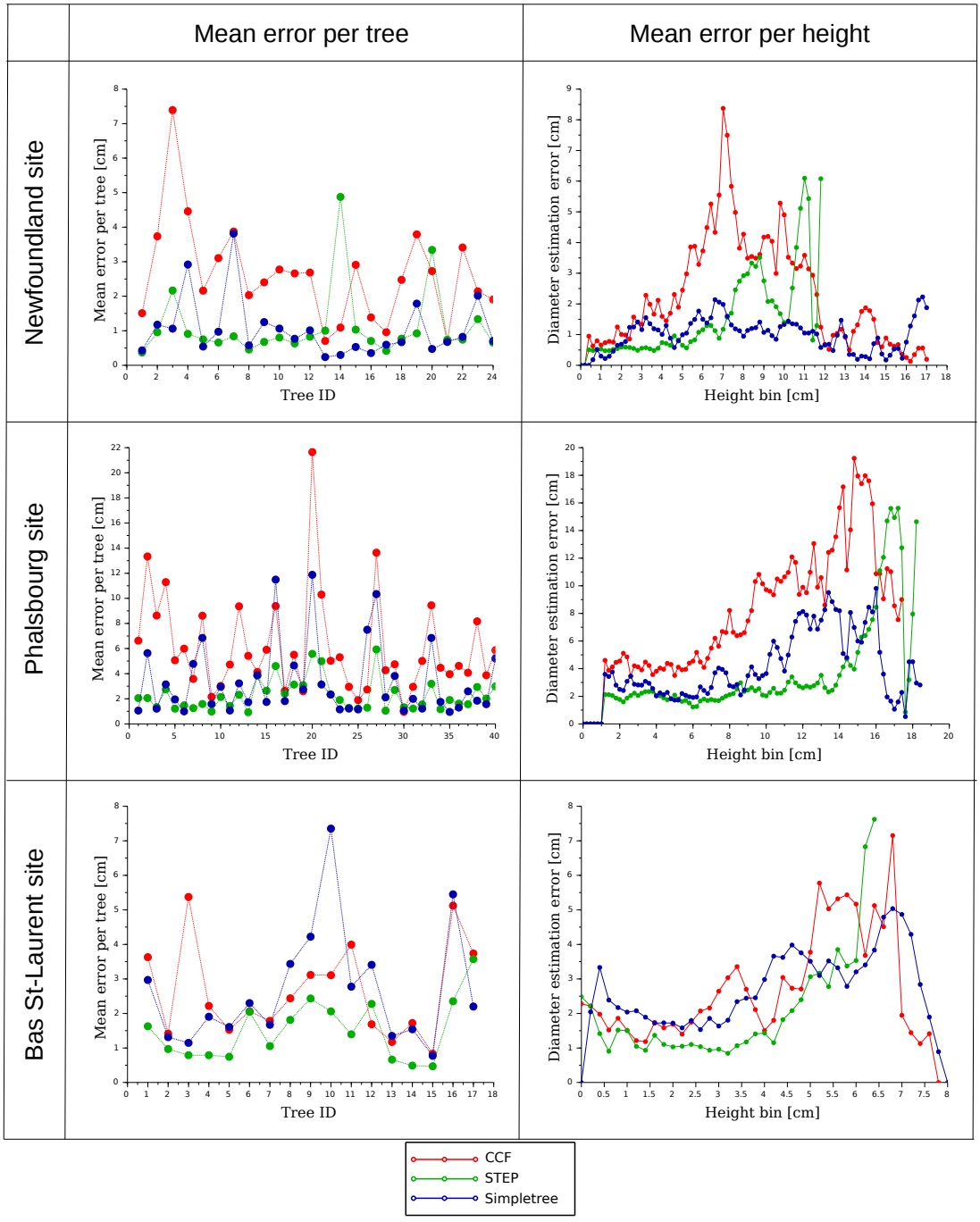


Figure V.8. – Diameter estimation error for each site and each algorithm.

6. Discussion

Our analysis allowed comparing three methods to estimate stem taper on three contrasted test sites. The comparison also benefited from reliable independent reference values on 81 trees that were either modelled, manually acquired, or provided by a scanner with finer resolution than the TLS (the True Shape unit). An obvious but important outcome from our results is the fact that stems are well identified by the three methods when there is little occlusion or noise. This emphasizes the importance to test the methods in real situation where occlusion and noise is predominant. Indeed, existing methods were mainly tested in open parks with clearly identifiable trees. However, forest inventories require to evaluate algorithms in real conditions as expected by operators.

The selection of an adapted algorithm depends on their accuracy and limitations. Our study highlighted several of them, including the height at which stem diameter can be estimated and the diameter estimation error. Note that interpretation of the results depend on two important aspects: Firstly, we restricted the seeds for the tuboid reconstruction with the STEP algorithm in the range of 1 m to 2 m. This affects the maximum height reconstruction of the algorithm since upper tuboids are intendedly ignored to avoid branches reconstruction. However, this limitation ensures that only tree stem is reconstructed which avoids the manual selection of cylinders required by the two other algorithms. Secondly we purposefully restricted any pre- and post-processing for the STEP algorithm to best test its resilience to deal with the occlusion and noise in the point cloud.

Several elements explain the difference of maximum estimation height reached by the three algorithms. The STEP algorithm is based on the tuboid growth which stops when no preferential growth direction is detected. This study revealed that two elements perturbed the tuboid growth. The high amount of noise in the upper part of trees from the the Bas Saint-Laurent site impacted accurate normal estimation. This affected the observation of normals convergence and the ensuing detection of growth direction. Trees from the Newfoundland site contained numerous branches, this resulted in several competing growth directions from which no clear preference was obtained. Setting more permissive growth stopping criterion should alleviate this issue. Note that considering tuboid seeds in the upper part of trees and accepting more than a single tuboid would also reconstruct tuboids at higher heights. The CCF algorithm considers horizontal layers of points, from bottom to top. Processing each layer individually allows to reconstruct separately fragments of stem in the upper part of trees, even if stem reconstruction failed in multiple layers. Therefore, even though reconstructed stems often contain several holes, the reconstruction height reaches upper part of the trees. These holes creates additional errors during diameter interpola-

tion and could be avoided by setting appropriate parameters that in turn would impact the maximum height reconstruction. The sphere following method included in SimpleTree is affected by occlusion and may stop before reaching tree tops. However, the latest version of the algorithm integrates a system of attractors combined with allometric relationships that adds cylinders reaching all unreconstructed parts of the point cloud. This system is the key to reconstruct tree tops but comes at the cost of additional errors. These results illustrate how finding a good balance between strict reconstruction and maximum height reconstruction. Therefore, making the algorithms self-reliant on parametrizing according to point cloud conditions is an important issue to consider for every solution.

Unlike for the STEP algorithm, the CCF and SimpleTree pipelines systematically applied different filters to maximize their results. This was equivalent to tests the brute abilities of the STEP algorithm to be a fully automated algorithm for stem taper estimate when compared with the best results we could extract from the CCF algorithm and SimpleTree. Consequently, it is predictable that the already favourable results for the STEP algorithm in most conditions would improve readily. For instance, applying common filters as standard pre-processing steps is recommended for all algorithms. A similar situation can be highlighted for algorithms post-processing. For instance, SimpleTree uses allometric relationships to correct some erroneous circles obtained by the sphere following procedure. Neither the STEP algorithm nor the CCF algorithm included such a correction. Therefore, inserting an equivalent correction could also improve the results of both algorithms. Our tests demonstrated that pre- and post-processing steps are important to efficiently improve the results.

The perspectives of automatic or semi-automatic algorithms, capable of estimating stem taper with high accuracy have never been so positive. Several factors contribute to accelerate their availability and their quick improvements to the point of being adopted in operational context of forest inventory. Firstly, they are becoming increasingly available in software platforms in an easy to run format. Our tests revealed the importance of the CompuTree platform along with convenience for objective comparisons of various point cloud processing algorithms. Other platforms or software interface are available but overall it is an important consideration to facilitate the use of complex algorithms applied to very large datasets. These user-friendly interfaces will also favour benchmarking, a crucial step towards improving existing methods by analysing their strengths and weaknesses. Secondly, several options are being explored for acquiring point clouds on a forest plots, and these may lead to technical advances in automated forest measurements. New instruments like UAV-mounted LiDARs (Unmanned Airborne Vehicles) may prove to be more adapted to estimate stem taper than TLS in diverse conditions. Thirdly, considering that pre- and post-processing su-

broutines can easily be implemented, the capacity of any given algorithm can be enhanced significantly. The encouraging results from the STEP algorithm of this study are good examples to illustrate how it can benefit from optional point cloud processing supplements readily available in the open source community. Lastly, we aim to select plot-level instead of tree-level point clouds for our next benchmarking of stem taper estimation algorithms to advance towards operational products supporting forest inventories. Such tests will help identifying how to parametrize efficiently the algorithms when confronted to various forest types. In the same time, the choice of normal estimation method will be further investigated to guide the application of the STEP algorithm to data sets with various characteristics.

7. Conclusion

We validated the STEP algorithm on a large set of trees extracted from TLS acquisition in forest environment under various conditions. Overall, the results shows that the accuracy of the STEP method is comparable to the accuracy of manual measurements with classical instruments, with mean errors in the range of 1 cm to 2 cm. We also compared the STEP algorithm to two other approaches dedicated to stem taper estimation. The results proved that the STEP algorithm is more adapted to occluded and noisy point clouds than the others. We highlighted the differences in the processing chains of each algorithm and showed that common pre- and post-processing could improve the current results. The increasing robustness of automatic measurement algorithms pushes the limits of LiDAR-aided forest inventories tools and new instruments open some interesting perspective for the use of the presented solutions in applicative contexts.

VI. Tree segmentation from tubular shapes

Overview

Terrestrial laser scanners (TLS) are able to acquire the geometry of trees with unrivalled precision. Therefore foresters have great expectations about TLS instruments and point cloud processing. The non-invasive analysis of trees geometry is the key to novel and accurate measurements. In particular, the cost of studying the upper part of trees is prohibiting, thus point cloud processing is an attractive solution for tree crowns measurements. The precise quantification of novel metrics related to tree crowns will improve the existing allometric and growth models, provide a better understanding of spatial interaction between trees and enhance our knowledge on forest ecosystems. This interest for tree crown measurements recently increased and generated a new need in forest TLS-scanned point cloud processing. Indeed, in order to measure tree crowns, trees have to be identified as separated objects. Therefore automatic segmentation of trees from TLS point clouds constitutes a important gain for improving significantly forest inventory procedures.

Several geometric challenges emanate from the automatic segmentation of trees inside a point cloud. Point clouds differ from meshes: the lack of adjacency between data points makes it difficult to apprehend the topology of a point set. Therefore, segmenting objects becomes a hard task. In addition, tree crowns highly suffer from large occlusions. Hence branches and foliage are not ”continuously“ sampled. Adaptive methods are hence required to identify separated clusters of points that belong to the same object. Statistical solutions cannot be applied efficiently due to the reduced sampling density in the upper part of trees and the non-homogeneity of the points. Therefore geometric approaches are preferred.

In this chapter we propose a solution to this problem and detail how we use geometric and morphologic information in point clouds to achieve the desired tree segmentation. In particular, we show how we reached our objective by first solving the segmentation on a volumetric representation of the point cloud by using principles taken from the graphs theory, before transcribing the results on the original data points.

1. Introduction

The measurements of upper parts of trees, and especially crowns shapes and dimensions are of major interest in forestry and botanic. These elements can not be accessed during field campaign. Therefore an emerging solution to fill this lack of data is to use TLS instruments which are able to capture the geometry of trees.

The first step for measuring tree crown attributes in a forest plot from a TLS point cloud is to assign each data point to a tree of the plot. This segmentation constitutes a major challenge in forest environment where tree crowns closely overlap and where the point cloud suffers from occlusion, noise and low resolution sampling. Therefore, point clouds are currently segmented by operators, either manually or using semi-automatic procedures [128]. Indeed, only a few automatic methods have been developed to reach this goal.

The work of [151] used the data points around detected tree stems to extract horizontal crown hulls from radial symmetry images. Currently, [53] are developing a method based on the distance from the points to the trees and their competitors. The authors included control polygons to support the resulting segmentation. Even though solutions emerge, forest point cloud processing faces TLS data limitations and validation issues. Indeed, these methods have not been validated yet since the lack of objective reference data makes the evaluation of tree crown segmentation difficult. Hence, similar to image processing (see the Berkeley reference data base [18]) human expertise is considered to be the actual reference for this problem [133].

In this context, our objective was to propose a novel methods for isolating the crown of each tree present in a TLS-scanned forest plot. The goals of this method was to cluster data points of each tree into a unique and coherent segment. The method was to be tested in natural forest environments with various inter-tree competition patterns in order to emphasize its limitations. It was validated against a reference manual isolation produced by an expert operator since it is consider to be the most reliable protocol to this day.

2. Data sets

In 2013, the UQAR (Université du Québec à Rimouski) conducted a TLS acquisition campaign near the towns of Squatec and Rimouski in the province of

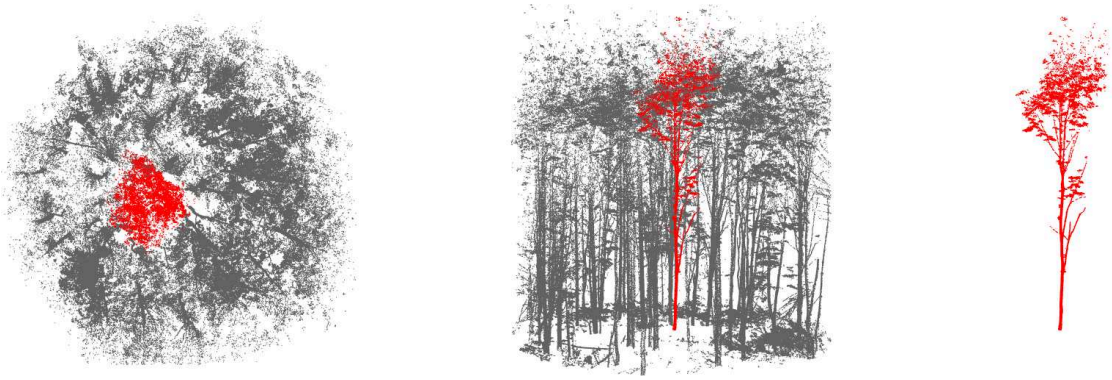


Figure VI.1. – Top view (left), side view (middle) and result (right) of a manual tree crown segmentation from a TLS point cloud. The tree in red has been isolated from its grey competitors.

Québec. This campaign aimed at studying competition between Sugar Maple and Balsam Firs. 192 target trees in competitive environment and their surrounding were scanned. Among them we selected 32 trees for the evaluation of our algorithm, corresponding to the 4 cases of species competition: 8 Maples surrounded by Maples, 8 Maples in competition with Firs, 8 Firs located inside Maple trees, and 8 Firs in a Maple stand. The point clouds were preprocessed to include only the competitive environment of each target tree (circular plot centred around the target tree with a radius depending on the height of the closest competitor). Each target tree was manually segmented in the resulting point clouds by a trained operator (Figure VI.1). Therefore we considered this segmentation as the reference measurement during the evaluation of our algorithm [133].

3. Methods

The automatic tree crown segmentation method we developed requires a previous estimation of trees location inside the TLS-scanned forest plot (which can be achieved using the STEP algorithm for example). Our segmentation is based on two characteristics of trees: on one hand trees are continuous connected objects, and on the other hand the vegetation density decreases where tree crowns overlap (Figure VI.2). This phenomenon is called *crown shyness*. It is an effect of the wind that generates frictions between trees which lead to branches breakages.

Our methodology is divided into three main steps (Figure VI.3). In a first step the input point cloud is discretized into voxels and connected components (CC)

Figure VI.2. – The crown shyness induces a diminution of vegetation density where crowns overlap, facilitating the segmentation of tree crowns.

are identified. During the second step, the resulting CC are divided until they they intersect at most one tree. Finally, a neighbourhood graph is computed in the third step to assign each isolated CC to a tree of the forest plot.

3.1. Identification of connected components

The first step of our method identifies connected objects in the 3D point cloud. We achieve this goal by first generating a 3D density grid on the point cloud: a regular 3D grid in which the intensity of each voxel is equal to the number of points it contains. Then, the density grid is segmented into CC considering a 26 neighbourhood (Figure VI.4c).

We fixed the grid resolution to 2 cm for this first step. This value is a compromise between the distance of two consecutive data points, the distance separating trees and the needs of computer resources. This way, two elements further than 4 cm apart will be separated into two distinct CC. However this step is insufficient since each CC does not represent a tree. Indeed, when crowns overlap, two or more trees may be merged into a single CC. Also, occlusions in the point clouds separates tree foliage into multiple CC.

3.2. Obtaining Simplex components

The second step of our method aims at finding a one-to-one relationship between the input location of trees in the forest plot and a subset of the identified

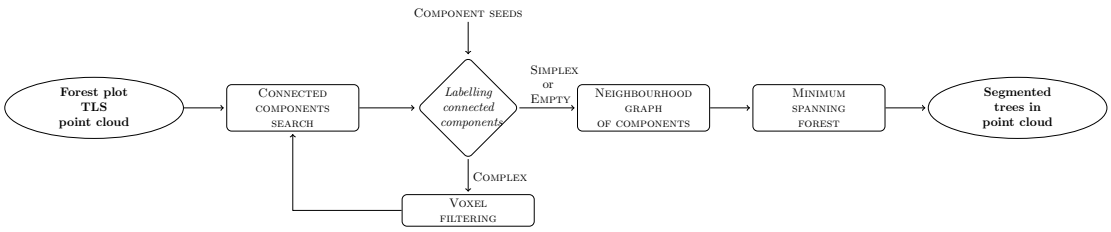


Figure VI.3. – Overview of the proposed methodology

CC. To do so, we assign a label to each CC depending on the number of tree location they intersect: components intersecting two or more trees are labelled as COMPLEX, component intersecting a single tree are labelled as SIMPLEX and components not intersecting trees are labelled as EMPTY (Figure VI.4d). To obtain a one-to-one relationship, COMPLEX components must then be split: we recursively divide each COMPLEX component C_i into a set of sub components $C_{i,j}$ until all resulting components are either SIMPLEX or EMPTY.

The division of a CC requires to remove voxels such that the component is not connected any more. COMPLEX components are the result of crown overlap. Therefore we take advantage of the crown shyness to divide them along the low vegetation density areas: We iteratively filter the voxels with an incremental threshold on the vegetation density (the value of the threshold increases at each iteration) until obtention of SIMPLEX and EMPTY components. This way, the first voxels filtered correspond to the areas of low vegetation density, thus to crown shyness. The incremental threshold ensures to filter the minimum number of voxels required to split the CC. This second step produces a set of SIMPLEX and EMPTY CC (Figure VI.4e). Nevertheless, while SIMPLEX CC correspond to a single tree, EMPTY CC have to be attributed back to a tree.

3.3. Graph creation and minimum spanning forest

In a third step, we assign each Empty CC to a Simplex CC (hence to a tree). Assuming that CC of the same tree are close one to another, we use the principle of nearest neighbour to achieve this task. As a starting point we compute a neighbourhood graph G of all the CC (Figure VI.4f): $G = (C, E)$ is an undirected graph in which each vertex $c \in C$ is a CC and each edge $e = (c_i, c_j) \in E$ represents a neighbour relationship between vertices. The edges of G are weighted according to the minimum distance separating the neighbours CC. I.e. the weight w_k of the edge $e_k = c_i, c_j$ is given by:

$$w_k = \arg\min\left(d(v_a, v_b)\right) \tag{VI.1}$$

where $v_a \in c_i$ is a voxel of the i^{th} CC, $v_b \in c_j$ a voxel of the j^{th} CC, and d is the Euclidean distance.

We then compute a minimum spanning forest F from the graph G and the Simplex CC (Figure VI.4g): The sum of the edges weights of the cycle-free subgraph F of G is minimal. Therefore this operation creates a graph in which the proximity between CC is maximised. Finally, Empty CC are assigned back to a Simplex CC using a breadth-first-search algorithm on F. Each CC is therefore assigned a unique tree identifier used to segment the data points (Figure VI.4b).

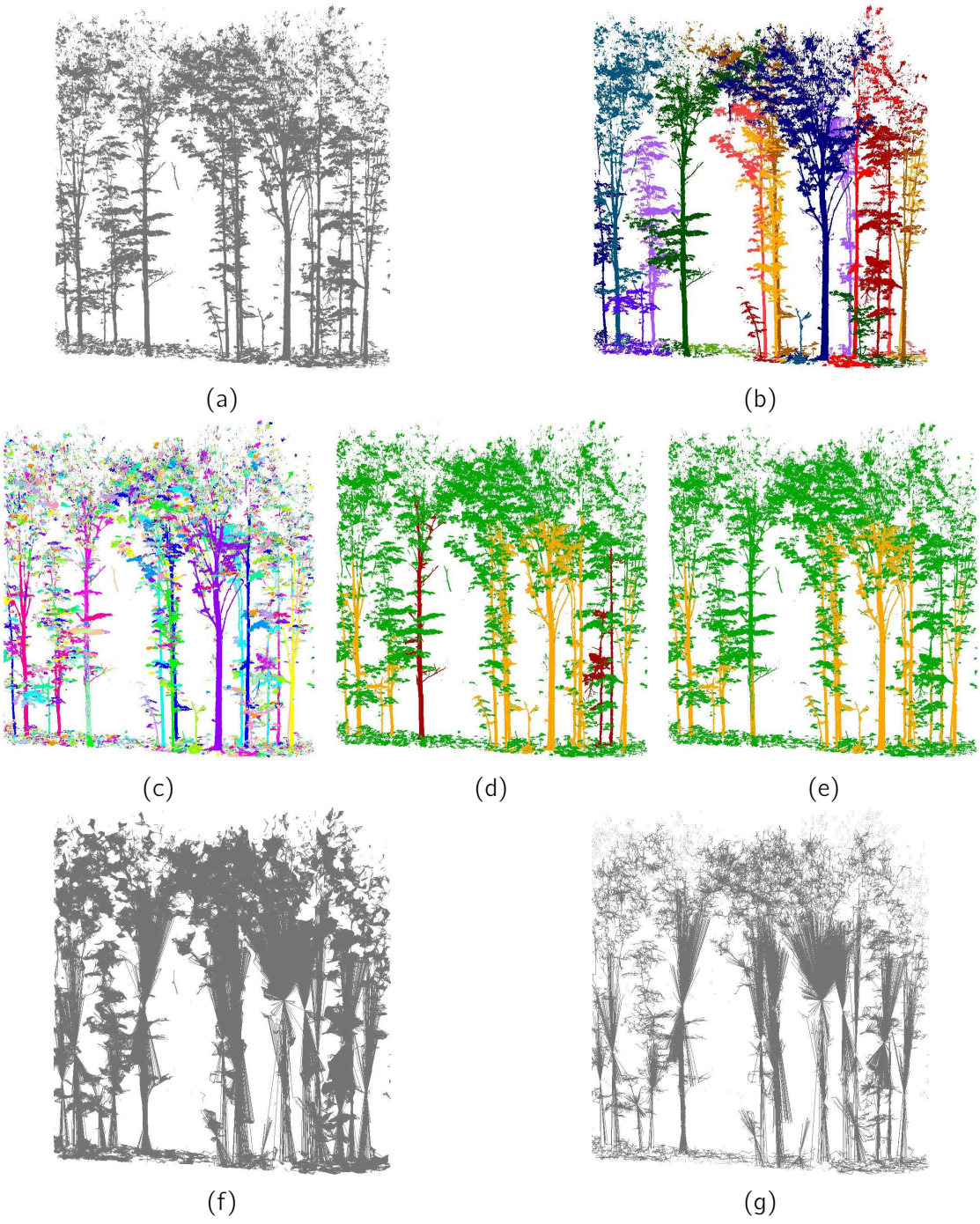
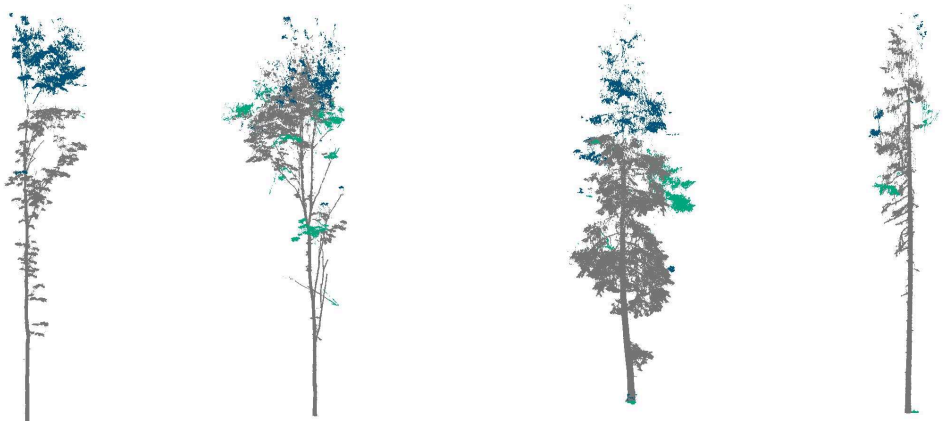


Figure VI.4. – Steps for segmenting a forest plot point cloud (a) into isolated trees (b). After the identification of connected components (c), they labelled as Complex (red), Simplex (yellow), or Empty (green) (d) and Complexes components are divided into Simplexes and Empty components. Finally, the computation of a neighbourhood graph (f) and its minimum spanning forest (g) generate the final segmentation.



(a) (b) (c) (d)

Figure VI.5. – Visualisation of commission (green) and omission (blue) errors for a Maple surrounded by Maples (a), a Maple surrounded by Firs (b), a Fir surrounded by Maples (c) and a Fir surrounded by Firs (d).

4. Preliminary results

We tested our automatic tree segmentation on the selected data sets. Target tree segmented by our algorithm were first identified and extracted. Then they were compared to the same target tree isolated manually by a trained operator in order to quantify the accuracy of our method (Figure VI.5). Table VI.1 summarized for each target tree the number of points correctly segmented (success rate), the number of points wrongly identified to another tree (omission error), the number of points wrongly identified to the target tree (commission error) and the total error between our method and the manual segmentation. Overall, the average total error was always below 30% with an average of 12.38% and a standard deviation of 9.6%. However results show a difference of accuracy depending on the type of environment: confronted to a plot made of deciduous trees, our method generates an average error around 17%, while the average error decreases to less than 7% in coniferous environments.

5. Discussion

The development of an automatic tree segmentation algorithm, including tree crowns, from TLS-scanned point clouds is an important contribution in the context of computer-aided forest inventories. Indeed, the automation of this procedure is a consistent gain since this problem was rarely addressed due to its complexity

Tree specie and competitor	Number of points	Success rate [%]	Omission rate [%]	Commission rate [%]	Total error [%]
Maple - Maple	967747	87.53	12.45	0.01	12.47
	1776321	70.35	28.70	0.95	29.65
Maple - Fir	2436491	89.41	2.38	8.21	10.59
	752606	81.51	5.75	12.75	18.49
Fir - Maple	1783455	97.27	1.38	1.365	2.73
	455888	81.19	2.72	16.09	18.81
Fir - Fir	1614432	97.02	0.79	2.19	2.98
	2905969	96.67	1.24	2.08	3.33

Table VI.1. – Accuracy of our automatic tree crown segmentation algorithm.

[133]. It gives a simplified access to various tree crown measurements and opens the gate to the computation of new forestry metrics [128].

The propose method differs from what is currently available: it considers neither a vertical continuity like in [151], nor a horizontal distance between data points and tree locations [53]. It relies on simple principles as the identification of connected components. The methods takes advantage of a 3D grid to compute efficiently the neighbour relationships and distances between components. The graph representation offers a simple representation of the data and diminishes the complexity of further processing.

The automatic method segmentation was tested on challenging data sets and preliminary results showed the limitations of the segmentation when facing deciduous environments. The proximity and the density of competitive branches partially explain the higher percentage of error. Also the presented method includes an additional weakness: even though the use of graphs facilitates the segmentation, it is the cause of incertitudes. Indeed, the unicity of a minimum spanning forest from a given graph is not ensured. Hence, it is possible that the minimum spanning forest used in the automatic segmentation does not lead to optimal accuracy. In the current state of our work, this problem remains to be solved.

6. Conclusion

In this study we presented a solution to an unresolved problem. Our method automatically segment the trees of a TLS-scanned forest plot based on an their approximative location. It gives accurate results (with error mostly under than

5%) when confronted to coniferous environments but needs to be improve to reach the same accuracy when applied on deciduous trees. However, this study is an ongoing work and we plan to implement consistent improvements. In addition, we tested our method on particularly challenging data since target trees were in strong competition with their environment, thus data contained a large amount of crown overlap. It is very likely our method would achieve higher accuracy when applied on less competitive environments.

Several elements are being examined to improve the quality of our method. We will first focus our efforts on solving the issue of non-unicity of the minimum spanning forest. We believe that appropriate heuristics and fuzzy approaches could greatly improve our results. Separately, we will aim at refining the division of COMPLEX connected components in order to improve the starting point of the graph processing step.

With enhanced results, automatic segmentation of trees could largely benefit to forestry by giving access to objective and precise new measurements unattainable until now.

Conclusion

Point cloud processing still faces major limitations. The most important of them is the lack of topological information (absence of connection between data points), occlusion and missing parts, noise and non-homogeneous sampling density. In this context, apprehending the geometry of point-sampled surfaces is a difficult task. The extraction of high level information on a target object is even more challenging.

In this thesis we investigated solutions to this issue and proposed a robust tubular shape extraction from point clouds. We intendedly chose to avoid preprocessing algorithms that reduce the presence of noise and outliers: these algorithms alter the geometry of the original point cloud which is unwanted in measurement applications like forestry or reverse-engineering. Our works provided high level information about the content of a noisy and occluded point cloud using geometric tools. Among several potential applications (e.g. industry monitoring), we chose to apply our method to forest data sets which highly suffer from point clouds limitations. Processing large forest point clouds gave rise to two distinct developments with reduced complexity: a fast normal vector estimation method, and an efficient tubular shape extraction. Based on these developments we also presented an object-oriented segmentation algorithm which aims at minimising the effects of the lack of topology along with occlusion and low sampling density.

Our researches provide solutions to geometric questions of general order. We investigated the possibilities to achieve fast method for differential geometry approximation on point clouds. We also studied a general pattern recognition issue that is: *how to extract a shape with an a priori about its geometry?* Particularly, we considered the means to detect and reconstruct specific surfaces from imperfect point clouds using adapted parametric models. We also investigated the possibilities to overcome the lack of connection between points while coherently segmenting connected objects in a point cloud.

In addition, we addressed more specific geometric issues. First, the anisotropic octrees provide a solution to model complex geometries with a collection of simple surface models. Therefore we reached a simpler representation of objects that may be easier to handle throughout different processing. Also we solved the tubular shape extraction problem using an innovative parametric model based on a 4D curve. And finally we considered fundamental geometry to make the active contours energy parametrization independent.

Several geometric choices precluded our works based on the available tools. We chose to use parametric models because they are well adapted for pattern recognition purposes, easy to handle and provide a fast access to differential geometry. The proposed fast normal estimation method indeed uses a point cloud representation build on a collection of quadratic elevation surfaces. Similarly, the tubular shape extraction method models the tubular shapes as parametric surfaces based on a 4D curve embedded in a parametric space. Finally we decided to use volumetric models, with 3D grid representation of point clouds, to overcome the lack of connection between points.

In order to achieve a fast normal vectors estimation method, we considered a balancing between results accuracy and computing time. The considerations resulted in anisotropic octrees dividing the point cloud into patches of points which size reflects the local complexity of the geometry. In addition to normal estimation, this data structure is the root of a point cloud compression algorithm and shares the same quality for neighbour finding as traditional octrees. Therefore, anisotropic octrees appear to be very attractive: they offer more possibilities than classical data structure at the cost of a small increase of the initialisation computing time. Moreover, the approximation of the point cloud geometry with quadratic surfaces stored in octree nodes also gives a curvature indicator which is not a trivial task on point clouds. At a higher level, this study reveals the need of adapted partitioning of point clouds, and that researches are still required to transpose existing efficient anisotropic processing on point cloud data.

We also focused on the robustness of shapes extraction method: contrary to existing general methods, our tubular shapes extraction algorithm was developed to be robust to severe data imperfections. The proposed algorithm is also very different from existing forestry-specialised algorithms because it reconstruct full and continuous tubular shapes rather than a stack of cylindrical or circular elements. We chose to follow an accumulation approach that tends to diminish the impact of noise on the final shape reconstruction. Moreover, the Hough transform has proven to be able to reconstruct full shapes from partial and incomplete data, hence we took advantage of this property to overcome point cloud occlusion. We also tackled the non-homogeneous sampling density with local analyses during open growing active contours growth. The results obtained on forest data sets proved the pertinence of our choices: we validated our methodological decisions and showed the performance of our development confronted to noisy and occluded raw point clouds. Our algorithm provides objective and automated measurements that reach the accuracy of manual inventories. Finally, our method has proven to outperform existing algorithms when confronted to noisy and occluded point clouds.

We also tested the viability of volumetric models for point cloud segmentation.

This representation benefits from implicit neighbour relationships and we used it to transpose a geometrical question to a simpler graph problem. Doing so, we offer a first solution for an object-oriented point cloud segmentation of complex scenes affected by occlusion and low sampling density. This work is crucial for efficient advances in forestry science since it facilitates the study of tree crowns and gives access to objective botanical measurements inaccessible until now.

Our work end up with the creation of a plug-in integrated to the Computree software platform. This plug-in includes a complete applicative pipeline processing adapted for forestry purposes in which each algorithm feeds the following in the pipeline sequence. We focused on forestry applications because they are well suited for validation purposes: forest point clouds suffer from additional imperfections, which revealed the strengths and weaknesses of our methods. However, it is important to note that the presented work is not restricted to forestry usages. Indeed, normal estimation is a very general issue in point cloud processing, and we do not assume prior forestry specific knowledge. Our algorithms can therefore be applied on every type of point cloud. Similarly, tubular shape extraction is a very general issue which finds other applications: in industry monitoring, for example, where controlling pipes condition is important.

The proposed methods bring new research topics in forestry, geometry and remote sensing applications. The very next steps in forest measurements will be to apply our methods at forest plot scale and integrate specific allometric rules in our developments. In the future we aim at using our algorithm for tubular shape extraction to provide a complete QSM model of trees competing with the state-of-the-art proposals. Also, with little additions to our work, we plan to support the efficient computation of wood volume and derive accurate tree architectural models from TLS acquisitions. Based on the automatic segmentation of individual trees from a forest plot, we propose to push the limits of current available measurements and estimate tree crown attributes such as width, height, total volume, Leaf Area Index (LAI) or envelope. These estimations could further lead to the proposition of interesting competitive indices between trees.

With further works, our methods could be improved. For example, with new subdivision schemes and stopping criteria, we believe that anisotropic octrees could run faster while maintaining the accuracy of the presented $RMSE$ criterion. Also, when the rigidity of elevation surface models, we plan to consider other geometric models to better approximate the point cloud. Also, we aim at proposing further algorithms (e.g. segmentation, smoothing) inspired by existing anisotropic processing on images or meshes. The tubular shape extraction algorithm includes several key elements to be improved. We plan to embed our work in a multi-scale analysis, and to focus on developing new approaches for the extraction of a curve of interest inside a discrete space. Similarly, the integra-

tion of curvature indicators appears to be attractive for the extraction of tubular shapes. The applications of the presented Hough transform variant offers interesting perspectives: (1) it shares common points with the signed distance transform and hence could be used to produce skeletons of general shapes, and (2) each primitive deriving from a surface of revolution (e.g. sphere, cylinder, cone and torus) has an identifiable signature in the Hough space, therefore efforts will be put to develop shape classification approaches. Finally, new distances, the inclusion of heuristics and fuzzy segmentations could readily improve the proposed tree segmentation algorithm.

With the increasing popularity of laser scanning technology and the variety of support for the sensors we aim at enlarging our field of application. In particular, the accuracy of UAV-mounted LiDARs is promising for forestry purposes. Demonstrating the potential of our algorithms on such data would reinforce their attractiveness. Point clouds acquired in industrial or urban environments do not suffer the same limitations as forest data sets, and the tubular shape extraction algorithm would greatly benefit to human-made structures control. Indeed, major sites such as nuclear plants, off-shore platforms or pipelines require a strict monitoring to ensure their security and stability. Laser scanners are powerful tools that come with new questions, and today the perspectives of application are very wide.

References

[1] Fabian AITEANU et Reinhard KLEIN. « Hybrid tree reconstruction from inhomogeneous point clouds ». In : *Visual Computer* 30.6-8 (2014), p. 763–771 (cf. p. 153).

[2] Markku ÅKERBLOM, Pasi RAUMONEN, Mikko KAASALAINEN et al. « Analysis of Geometric Primitives in Quantitative Structure Models of Tree Stems ». In : *Remote Sens* 7 (2015), p. 4581–4603 (cf. p. 153).

[3] N AKKIRAJU, H EDELSBRUNNER, M FACELLO et al. *Alpha shapes : definition and software*. 1995 (cf. p. 86).

[4] Marc ALEXA, Johannes BEHR, Daniel COHEN-OR et al. « Computing and rendering point set surfaces ». In : *IEEE Transactions on Visualization and Computer Graphics* 9.1 (2003), p. 3–15 (cf. p. 42, 56, 71, 73).

[5] John AMANATIDES et Andrew WOO. « A Fast Voxel Traversal Algorithm for Ray Tracing ». In : *Proceedings of EUROGRAPHICS*. T. 87. 3. 1987, p. 3–10 (cf. p. 130).

[6] N AMENTA et M BERN. « Surface reconstruction by Voronoi filtering ». In : *Discrete & Computational Geometry* 22.4 (1999), p. 481–504 (cf. p. 64).

[7] Nina AMENTA, Marshall BERN et Manolis KAMVYSSELIS. « A new Voronoi-based surface reconstruction algorithm ». In : *Proceedings of the 25th annual conference on Computer graphics and interactive techniques - SIGGRAPH '98* 32 (1998), p. 415–421 (cf. p. 86).

[8] T ASCHOFF et H SPIECKER. « Algorithms for the automatic detection of trees in laser scanner data ». In : *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 36 (Part 8.W2 (2004), p. 71–75 (cf. p. 153).

[9] T ASCHOFF, M THIES et H SPIECKER. « Describing Forest Stands Using Terrestrial Laser-Scanning ». In : *ISPRS-International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* XXXV.B (2004), p. 237–241 (cf. p. 153).

[10] T E AVERY et H E BURKHART. *Forest Measurements*. McGraw-Hill series in forest resources. McGraw-Hill, 2001 (cf. p. 24, 151).

[11] D. H. BALLARD. « Generalizing the Hough transform to detect arbitrary shapes ». In : *Pattern Recognition* 13.2 (1980), p. 111–122 (cf. p. 97).

[12] Martin BÉLAND, Dennis D. BALDOCCHI, Jean Luc WIDLOWSKI et al. « On seeing the wood from the leaves and the role of voxel size in determining leaf area distribution of forests with terrestrial LiDAR ». In : *Agricultural and Forest Meteorology* 184 (2014), p. 82–97 (cf. p. 58, 78).

[13] Martin BÉLAND, Jean-Luc WIDLOWSKI, Richard A FOURNIER et al. « Estimating leaf area distribution in savanna trees from terrestrial LiDAR measurements ». In : *Agricultural and Forest Meteorology* 151.9 (2011), p. 1252–1266 (cf. p. 152).

[14] Ben BELLEKENS, Vincent SPRUYT et Maarten WEYN. « A Survey of Rigid 3D Pointcloud Registration Algorithms ». In : *AMBIENT 2014, The Fourth International Conference on Ambient Computing, Applications, Services and Technologies. 2014.* c (2014), p. 8–13 (cf. p. 68, 69).

[15] D. BELTON, S. MONCRIEFF et J. CHAPMAN. « Processing tree point clouds using Gaussian Mixture Models ». In : *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* II-5/W2.November (2013), p. 43–48 (cf. p. 124, 153).

[16] Marcel BERGER et Bernard GOSTIAUX. *Differential Geometry : Manifolds, Curves, and Surfaces : Manifolds, Curves, and Surfaces*. T. 115. Springer Science & Business Media, 2012 (cf. p. 63).

[17] Matthew BERGER, Andrea TAGLIASACCHI, Lee M SEVERSKY et al. « State of the Art in Surface Reconstruction from Point Clouds ». In : *Proceedings of the Eurographics 2014, Eurographics STARs* 1.1 (2014), p. 161–185 (cf. p. 40, 50, 85).

[18] BERKELEY. *The Berkeley Segmentation Dataset and Benchmark* (cf. p. 173).

[19] Fausto BERNARDINI, Joshua MITTLEMAN, Holly RUSHMEIER et al. « The ball-pivoting algorithm for surface reconstruction ». In : *IEEE Transactions on Visualization and Computer Graphics* 5.4 (1999), p. 349–359 (cf. p. 86, 88).

[20] P.J. BESL et H.D. MCKAY. « A method for registration of 3-D shapes ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (1992), p. 239–256 (cf. p. 69).

[21] a BIENERT, Hg MAAS et S SCHELLER. « Analysis of the information content of terrestrial laserscanner point clouds for the automatic determination of forest inventory parameters ». In : *Workshop on 3D Remote Sensing in Forestry* (2006), p. 1–7 (cf. p. 124).

[22] A BIENERT, S SCHELLER, E KEANE et al. « Application of Terrestrial Laser Scanners For The Determination Of Forest Inventory Parameters. » In : *International archives of Photogrammetry , Remote Sensing and Spatial Information Science* 36 (2006), p. 5. arXiv : b (cf. p. 153).

[23] Anne BIENERT et Hans-gerd MAAS. « Methods for the Automatic Geometric Registration of Terrestrial Laser Scanner Point Clouds in Forest Stands ». In : *Laserscanning 2009* XXXVIII (2009), p. 93–98 (cf. p. 69).

[24] Danny BLANCHETTE, Richard A FOURNIER, Joan E LUTHER et al. « Forest Ecology and Management Predicting wood fiber attributes using local-scale metrics from terrestrial LiDAR data : A case study of Newfoundland conifer species ». In : *Forest Ecology and Management* 347 (2015), p. 116–129 (cf. p. 152).

[25] Harry BLUM. « A Transformation for Extracting New Descriptors of Shape ». In : *Models for the Perception of Speech and Visual Form*. Sous la dir. de Weiant WATHEN-DUNN. Cambridge : MIT Press, 1967, p. 362–380 (cf. p. 80).

[26] Jan BÖHM et Susanne BECKER. « Automatic Marker-Free Registration of Terrestrial Laser Scans Using Reflectance Features ». In : *Proceedings of 8th Conference on Optical 3D Measurement Techniques* 2005 (2007), p. 338–344 (cf. p. 69).

[27] Dorit BORRMANN, Jan ELSEBERG, Kai LINGEMANN et al. « The 3D Hough Transform for plane detection in point clouds : A review and a new accumulator design ». In : *3D Research* 2.2 (2011), p. 1–13 (cf. p. 51, 97, 98).

[28] Mario BOTSCH, Leif KOBBELT, Mark PAULY et al. *Polygon mesh processing*. CRC press, 2010 (cf. p. 45, 46).

[29] Alexandre BOULCH et Renaud MARLET. « Fast and Robust Normal Estimation for Point Clouds with Sharp Features ». In : *Computer Graphics Forum* 31.5 (août 2012), p. 1765–1774 (cf. p. 64).

[30] Alexandre BOULCH et Renaud MARLET. « Deep Learning for Robust Normal Estimation in Unstructured Point Clouds ». In : *Computer Graphics Forum* 35.5 (2016), p. 281–290 (cf. p. 64).

[31] M BOUVIER, S DURRIEU, R FOURNIER et al. « Influence of sampling design parameters on biomass predictions derived from airborne lidar data ». In : *SilviLaser 2015*. La Grande Motte, France, 2015, p. 137–139 (cf. p. 152).

[32] G BROLLY et G KIRALY. « Algorithms for stem mapping by means of terrestrial laser scanning. » In : *Acta Silvatica et Lignaria Hungarica* 5 (2009), p. 119–130 (cf. p. 77, 124).

[33] Andreas BRUNNER et Belachew GIZACHEW. « Rapid detection of stand density, tree positions, and tree diameter with a 2D terrestrial laser scanner ». In : *European Journal of Forest Research* 133.5 (2014), p. 819–831 (cf. p. 124, 153).

[34] A. BUCKSCH, R.C. LINDENBERGH et M. MENENTI. « SkelTre - Fast skeletonisation for imperfect point cloud data of botanic trees ». In : *Eurographics Workshop on 3D Object Retrieval, EG 3DOR*. 2009, p. 13–20 (cf. p. 83).

[35] Alexander BUCKSCH et Roderik LINDENBERGH. « CAMPINO - A skeletonization method for point cloud processing ». In : *ISPRS Journal of Photogrammetry and Remote Sensing* 63.1 (2008), p. 115–127 (cf. p. 83).

[36] Martin D BUHMANN. *Radial basis functions : theory and implementations*. T. 12. Cambridge university press, 2003 (cf. p. 49, 90).

[37] J. C. CARR, R. K. BEATSON, J. B. CHERRIE et al. « Reconstruction and Representation of 3D Objects with Radial Basis Functions ». In : *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), p. 67–76. arXiv : ACMPress,NewYork,NY,USA (cf. p. 49, 85, 90–92).

[38] William CHANG. *Surface reconstruction : from points to splines*. San Diego, 2008 (cf. p. 85).

[39] Qi CHEN, Dennis BALDOCCHI, Peng GONG et al. « Isolating individual trees in a savanna woodland using small footprint lidar data ». In : *Photogrammetric Engineering & Remote Sensing* 72.8 (2006), p. 923–932 (cf. p. 78).

[40] Siheng CHEN, Dong TIAN, Chen FENG et al. « Fast Resampling of 3D Point Clouds via Graphs ». In : *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on* June (2016), p. 1–13. arXiv : 1702.06397 (cf. p. 75).

[41] Siu-Wing CHENG, Tamal K DEY et Jonathan SHEWCHUK. *Delaunay mesh generation*. CRC Press, 2012 (cf. p. 46, 86).

[42] Laurent D. COHEN. « On active contour models and balloons ». In : *CVGIP : Image Understanding* 53.2 (1991), p. 211–218 (cf. p. 126).

[43] « Constructive Solid Geometry ». In : *Introduction to Geometric Computing*. London : Springer London, 2008, p. 277–283 (cf. p. 50).

[44] Jean François CÔTÉ, Richard A. FOURNIER et Richard EGLI. « An architectural model of trees to estimate forest structural attributes using terrestrial LiDAR ». In : *Environmental Modelling and Software* 26.6 (2011), p. 761–777 (cf. p. 83, 152, 154, 158).

[45] Jean François CÔTÉ, Richard A. FOURNIER, Gordon W. FRAZER et al. « A fine-scale architectural model of trees to enhance LiDAR-derived measurements of forest canopy structure ». In : *Agricultural and Forest Meteorology* 166-167 (2012), p. 72–85 (cf. p. 152, 154, 158).

[46] Jean François CÔTÉ, Richard A. FOURNIER et Joan E. LUTHER. « Validation of L-Architect model for balsam fir and black spruce trees with structural measurements ». In : *Canadian Journal of Remote Sensing* 39.SUPPL.1 (2013) (cf. p. 154, 158).

[47] Mathieu DASSOT, Thiéry CONSTANT et Meriem FOURNIER. « The use of terrestrial LiDAR technology in forest science : Application fields, benefits and challenges ». In : *Annals of Forest Science* 68.5 (juil. 2011), p. 959–974 (cf. p. 124, 152).

[48] ONF R&D DÉPARTEMENT. *Computree official site*. 2017 (cf. p. 155).

[49] Jean-Emmanuel DESCHAUD et François GOULETTE. « Point Cloud Non Local Denoising using Local Surface Descriptor Similarity ». In : *PCV 2020 IISPRS Commission III Symposium Photogrammetric Computer Vision and Image Analysis* XXXVIII (2010), p. 109–114 (cf. p. 71).

[50] T.K. K DEY, G. LI et J. SUN. « Normal estimation for point clouds : a comparison study for a Voronoi based method ». In : *Proceedings Eurographics/IEEE VGTC Symposium Point-Based Graphics, 2005*. Ieee, 2005, p. 39–46 (cf. p. 64).

[51] Direction des inventaires forestiers (DIF) et Ministère des Forêts de la Faune et des Parcs (MFFP). *Norme d'inventaire écoforestier : PLACETTES-ÉCHANTILLONS TEMPORAIRES*. Rapp. tech. 2016, p. 1–172 (cf. p. 24, 25).

[52] Bertrand DOUILLARD, J. UNDERWOOD, N. KUNTZ et al. « On the segmentation of 3D LIDAR point clouds ». In : *International Conference on Robotics and Automation*. Ieee, mai 2011, p. 2798–2805. arXiv : ICRA.2011.5979818 [10.1109] (cf. p. 56, 58).

[53] Emmanuel DUCHATEAU. *Isolation d'arbres au sein de nuages de points LiDAR*. 2016 (cf. p. 173, 180).

[54] R DUMITRU, D BORRMANN et A NUCHTER. « Interior Reconstruction Using the 3D Hough Transform ». In : *3D-Arch 2013* XL.February (2013), p. 25–26 (cf. p. 99).

[55] Sylvie DURRIEU, T ALLOUIS, Richard FOURNIER et al. « Spatial quantification of vegetation density from terrestrial laser scanner data for characterization of 3D forest structure at plot level ». In : *SilviLaser*. 2004. 2008, p. 325–334 (cf. p. 58, 152).

[56] Herbert EDELSBRUNNER et Ernst P MÜCKE. « Three-Dimensional Alpha Shapes ». In : *ACM Transactions on Graphics* 13.1 (1994), p. 43–72. arXiv : 9410208 [math] (cf. p. 86).

[57] Jan ELSEBERG, Dorit BORRMANN et Andreas NÜCHTER. « One billion points in the cloud – an octree for efficient processing of 3D laser scans ». In : *ISPRS Journal of Photogrammetry and Remote Sensing* 76 (fév. 2013), p. 76–88 (cf. p. 60).

[58] Jan ELSEBERG, Stéphane MAGNENAT, Roland SIEGWART et al. « Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration ». In : *Journal of Software Engineering for Robotics (JOSER)* 3.1 (2012), p. 2–12 (cf. p. 60, 61).

[59] Michael J. FALKOWSKI, Alistair M S SMITH, Andrew T. HUDAK et al. « Automated estimation of individual conifer tree height and crown diameter via two-dimensional spatial wavelet analysis of lidar data ». In : *Canadian Journal of Remote Sensing*. T. 32. 2. 2006, p. 153–161 (cf. p. 152).

[60] Gerald FARIN. *Curves and surfaces for computer-aided geometric design : a practical guide*. Elsevier, 2014 (cf. p. 44).

[61] T. R. FELDPAUSCH, L. BANIN, O. L. PHILLIPS et al. « Height-diameter allometry of tropical forest trees ». In : *Biogeosciences* 8.5 (2011), p. 1081–1106 (cf. p. 151).

[62] Sagi FILIN. « Surface clustering from airborne laser scanning data ». In : *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences* 34 (2002), p. 119–124 (cf. p. 78).

[63] Martin a. FISCHLER et Robert C. BOLLES. « Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography ». In : *Communications of the ACM* 24.6 (1981), p. 381–395 (cf. p. 42, 96).

[64] Shachar FLEISHMAN, Iddo DRORI et Daniel COHEN-OR. « Bilateral mesh denoising ». In : *ACM Transactions on Graphics* 22.3 (2003), p. 950 (cf. p. 71).

[65] A FORTIN. *Analyse num{é}rique pour ing{é}nieurs*. Presses internationales Polytechnique, 2008 (cf. p. 42).

[66] Mathieu FORTIN, Robert SCHNEIDER et Jean Pierre SAUCIER. « Volume and error variance estimation using integrated stem taper models ». In : *Forest Science* 59.3 (2013), p. 345–358 (cf. p. 23, 151).

[67] B. GORTE. « Skeletonization of laser-scanned trees in the 3d raster domain ». In : *Innovations in 3D Geo Information Systems*. 2006, p. 371–380 (cf. p. 81).

[68] Ben H. GORTE et Norbert PFEIFER. « Structuring laser-scanned trees using 3D mathematical morphology ». In : *International Archives of Photogrammetry and Remote Sensing*. T. 35. 2004, p. 929–933 (cf. p. 81).

[69] Adrien GRESSIN, Clément MALLET, Jér Ôme DEMANTKÉ et al. « Towards 3D lidar point cloud registration improvement using optimal neighborhood knowledge ». In : *ISPRS Journal of Photogrammetry and Remote Sensing* 79 (2013), p. 240–251 (cf. p. 69).

[70] Stefan GUMHOLD, Xinlong WANG, Rob MACLEOD et al. « Feature extraction from point clouds ». In : *Proceedings of 10th international meshing roundtable*. 2001, p. 293–305 (cf. p. 53, 64).

[71] Jan HACKENBERG. *SimpleTree opendata page*. 2016 (cf. p. 143).

[72] Jan HACKENBERG, Christopher MORHART, Jonathan SHEPPARD et al. « Highly accurate tree models derived from terrestrial laser scan data : A method description ». In : *Forests* 5.5 (2014), p. 1069–1105 (cf. p. 154).

[73] Jan HACKENBERG, Heinrich SPIECKER, Kim CALDERS et al. « SimpleTree - An efficient open source tool to build tree models from TLS clouds ». In : *Forests* 6.11 (2015), p. 4245–4294 (cf. p. 154, 155).

[74] Jan HACKENBERG, Marc WASSENBERG, Heinrich SPIECKER et al. « Non Destructive Method for Biomass Prediction Combining TLS Derived Tree Volume and Wood Density ». In : *Forests* 6.4 (2015), p. 1274–1300 (cf. p. 145, 154).

[75] Xian Feng HAN, Jesse S. JIN, Ming Jie WANG et al. « A review of algorithms for filtering the 3D point cloud ». In : *Signal Processing : Image Communication* 57 (2017), p. 103–112 (cf. p. 71).

[76] M. C. HANSEN, P. V. POTAPOV, R. MOORE et al. « High-Resolution Global Maps of 21st-Century Forest Cover Change ». In : *Science* 342.6160 (2013), p. 850–853. arXiv : 1011.1669v3 (cf. p. 27).

[77] John C. HART. « Ray tracing implicit surfaces ». In : *SIGGRAPH '93 Course Notes : Design, Visualization and Animation of Implicit Surfaces* 1.4 (1993) (cf. p. 49).

[78] Jason G. HENNING et Philip J. RADTKE. « Detailed stem measurements of standing trees from ground-based scanning lidar ». In : *Forest Science* 52.1 (2006), p. 67–80 (cf. p. 153).

[79] Rüdiger HILDEBRANDT et André IOST. « From points to numbers : A database-driven approach to convert terrestrial LiDAR point clouds to tree volumes ». In : *European Journal of Forest Research* 131.6 (2012), p. 1857–1867 (cf. p. 125, 153).

[80] Shelley A HINSLEY, Ross A HILL, Paul E BELLAMY et al. « The application of lidar in woodland bird ecology : climate, canopy structure, and habitat quality ». In : *Photogrammetric Engineering and Remote Sensing* 72.12 (2006), p. 1399–1406 (cf. p. 23).

[81] Hugues HOPPE, Tony DEROSE, Tom DUCHAMP et al. « Surface reconstruction from unorganized points ». In : *SIGGRAPH Computer Graphics*. T. 26. 2. Juil. 1992, p. 71–78 (cf. p. 48, 53, 55, 56, 62, 64, 90).

[82] Hugues HOPPE, Tony DEROSE, Tom DUCHAMP et al. « Piecewise smooth surface reconstruction ». In : *Proceedings of the 21st annual conference on Computer graphics and interactive techniques - SIGGRAPH '94* (1994), p. 295–302 (cf. p. 87).

[83] F. HOSOI, Y. NAKAI et K. OMASA. « Voxel tree modeling for estimating leaf area density and woody material volume using 3-D LIDAR data ». In : *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* II-5.W2 (2013), p. 115–120 (cf. p. 152).

[84] PVC Paul V C HOUGH. *Method and means for recognizing complex patterns*. 1962 (cf. p. 97).

[85] H HUANG, Z LI, P GONG et al. « Automated methods for measuring DBH and tree heights with a commercial scanning lidar ». In : *Photogrammetric Engineering and Remote Sensing* 77.3 (2011), p. 219–227 (cf. p. 153).

[86] Julian S HUXLEY et Georges TEISSIER. « Terminology of relative growth ». In : *Nature* 137.3471 (1936), p. 780–781 (cf. p. 23).

[87] J ILLINGWORTH et J KITTLER. « SURVEY A Survey of the Hough Transform ». In : 44 (1988), p. 87–116 (cf. p. 97).

[88] INSTITUT NATIONAL DE L'INFORMATION GÉOGRAPHIQUE ET FORESTIÈRE. *IGN : inventaire forestier*. 2017 (cf. p. 24).

[89] Shuangshuang JIN, Robert R LEWIS et David WEST. « A comparison of algorithms for vertex normal computation ». In : *The Visual Computer* 21.1-2 (2005), p. 71–82 (cf. p. 64).

[90] Evangelos KALOGERAKIS, Derek NOWROUZEZAHRAI, Patricio SIMARI et al. « Extracting lines of curvature from noisy point clouds ». In : *Computer-Aided Design* 41.4 (avr. 2009), p. 282–292 (cf. p. 65).

[91] Eui Chul KANG, Duck Bong KIM et Kwan H. LEE. « Balanced feature-sensitive point sampling for 3D model generation ». In : *International Journal of Advanced Manufacturing Technology* 38.1-2 (2008), p. 130–142 (cf. p. 73, 74).

[92] Michael KASS, Andrew WITKIN et Demetri TERZOPOULOS. « Snakes : Active contour models ». In : *International Journal of Computer Vision* 1.4 (1988), p. 321–331 (cf. p. 126, 135).

[93] Michael KAZHDAN, Matthew BOLITHO et Hugues HOPPE. « Poisson Surface Reconstruction ». In : *Proceedings of the Symposium on Geometry Processing* (2006), p. 61–70. arXiv : 1006.4903 (cf. p. 53, 60, 62, 89, 92, 93).

[94] Dave KELBE, Paul ROMANCZYK, Jan van AARDT et al. « Reconstruction of 3D tree stem models from low-cost terrestrial laser scanner data ». In : *SPIE Defense, Security, and Sensing* 8731 (2013), p. 873106 (cf. p. 153).

[95] David KELBE et Paul ROMANCZYK. « Automatic extraction of tree stem models from single terrestrial lidar scans in structurally heterogeneous forest environments ». In : *Proceedings of Silvilaser* September (2012), p. 1–8 (cf. p. 124, 153).

[96] B KERAUTRET, A KRÄHENBÜHL, I DEBLED-RENNESSON et al. « 3D Geometric Analysis of Tubular Objects Based on Surface Normal Accumulation ». In : *Proc. Image Analysis and Processing (ICIAP 2015), Genova, Italy* 9279 (2015), p. 319–331. arXiv : 1506.06636 (cf. p. 62, 82, 83, 123).

[97] Carolyn KIMME, Dana BALLARD et Jack SKLANSKY. « Finding circles by an array of accumulators ». In : *Communications of the ACM* 18.2 (1975), p. 120–122 (cf. p. 127).

[98] G KIRÁLY et G BROLLY. « Tree Height Estimation Methods for Terrestrial Laser Scanning in a Forest Reserve ». In : *Proceedings of ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007*. 2007, p. 211–215 (cf. p. 153).

[99] K. KLASING, D. ALTHOFF, D. WOLLHERR et al. « Comparison of surface normal estimation methods for range sensing applications ». In : *IEEE International Conference on Robotics and Automation*. Ieee, mai 2009, p. 3206–3211 (cf. p. 64, 127).

[100] Michael KÖHL, Steen MAGNUSSEN et Marco MARCHETTI. *Sampling Methods, Remote Sensing and GIS Multiresource Forest Inventory*. T. 3. 2-3. 2006, p. 91–156 (cf. p. 24, 151).

[101] Dimitri KUDELSKI, Sophie VISEUR et Jean-Luc MARI. « Skeleton extraction of vertex sets lying on arbitrary triangulated 3D meshes ». In : *International Conference on Discrete Geometry for Computer Imagery*. Springer. 2013, p. 203–214 (cf. p. 65).

[102] Andres KUUSK, Mait LANG, Silja MÄRDLA et al. « Tree stems from terrestrial laser scanner measurements ». In : *Forestry Studies* 63.1 (2015), p. 44–55 (cf. p. 124, 153).

[103] Anthonie van LAAR et Alparslan AKCA. *Forest Mensuration*. T. 24. 624. 2007, p. 760–761. arXiv : arXiv:1011.1669v3 (cf. p. 24, 151).

[104] Sven LANGE, Niko SÜNDERHAUF, Peer NEUBERT et al. « Autonomous corridor flight of a UAV using a low-cost and light-weight RGB-D camera ». In : *Advances in Autonomous Mini Robots - Proceedings of the 6th AMiRE Symposium, AMiRE 2011*. 2012, p. 183–192 (cf. p. 74).

[105] Z LARI, A F HABIB et E KWAK. « An Adaptive Approach for Segmentation of 3D Laser Point Cloud ». In : *International Archives of the Photogrammetry* XXXVIII.August (2011), p. 29–31 (cf. p. 56).

[106] Guillaume LAVOUÉ, Florent DUPONT et Atilla BASKURT. « A new CAD mesh segmentation method, based on curvature tensor analysis ». In : *CAD Computer Aided Design* 37.10 (2005), p. 975–987 (cf. p. 65).

[107] E LEAL et Nalling LEAL. « Point cloud denoising using robust principal component analysis ». In : *Conference on Computer Graphics Theory and ...* January (2006), p. 51–58 (cf. p. 71).

[108] Nallig LEAL, Esmeide LEAL et John William BRANCH. « Simple method for constructing NURBS surfaces from unorganized points ». In : *Proceedings of the 19th International Meshing Roundtable, IMR 2010*. 2010, p. 161–175 (cf. p. 44, 88).

[109] K. H. LEE, H. WOO et T. SUK. « Point Data Reduction Using 3D Grids ». In : *The International Journal of Advanced Manufacturing Technology* 18.3 (2001), p. 201–210 (cf. p. 74).

[110] Pai Feng LEE et Bin Shyan JONG. « Point-based simplification algorithm ». In : *WSEAS Transactions on Computer Research* 3.1 (2008), p. 61–66 (cf. p. 65, 73).

[111] Guo LI, Ligang LIU, Hanlin ZHENG et al. « Analysis, reconstruction and manipulation using arterial snakes ». In : *ACM Transactions on Graphics*. 2010, 152 :1–152 :10 (cf. p. 122).

[112] Hua LI et Anthony YEZZI. « Vessels as 4-D curves : Global minimal 4-D paths to extract 3-D tubular surfaces and centerlines ». In : *IEEE transactions on medical imaging* 26.9 (2007), p. 1213–1223 (cf. p. 122).

[113] Yangyan LI, Xiaokun WU, Yiorgos CHRYSATHOU et al. « GlobFit : Consistently Fitting Primitives by Discovering Global Relations ». In : *ACM Trans. Graph.* 30.4 (2011), 52 :1–52 :12 (cf. p. 51, 53, 97).

[114] Xinlian LIANG, Ville KANKARE, Juha HYYPPÄ et al. « Terrestrial laser scanning in forest inventories ». In : *ISPRS Journal of Photogrammetry and Remote Sensing* 115 (2016), p. 63–77 (cf. p. 124, 152, 153, 155).

[115] Xinlian LIANG, Ville KANKARE, Xiaowei YU et al. « Automated stem curve measurement using terrestrial laser scanning ». In : *IEEE Transactions on Geoscience and Remote Sensing* 52.3 (2014), p. 1739–1748 (cf. p. 153).

[116] Xinlian LIANG, Paula LITKEY, Juha HYYPPÄ et al. « Automatic stem mapping using single-scan terrestrial laser scanning ». In : *IEEE Transactions on Geoscience and Remote Sensing* 50.2 (2012), p. 661–670 (cf. p. 153).

[117] Hong Wei LIN, Chiew Lan TAI et Guo Jin WANG. « A mesh reconstruction algorithm driven by an intrinsic property of a point cloud ». In : *CAD Computer Aided Design* 36.1 (2004), p. 1–9 (cf. p. 86).

[118] Eva LINDBERG, Johan HOLMGREN, Kenneth OLOFSSON et al. « Estimation of stem attributes using a combination of terrestrial and airborne laser scanning ». In : *European Journal of Forest Research* 131.6 (2012), p. 1917–1931 (cf. p. 153).

[119] L. LINSEN. « Point cloud representation ». In : *University of Karlsruhe, Germany Technical Report, Faculty of Informatics* (2001), p. 1–18 (cf. p. 73).

[120] Yu LIU et Youlun XIONG. « Automatic segmentation of unorganized noisy point clouds based on the Gaussian map ». In : *Computer-Aided Design* 40.5 (mai 2008), p. 576–594 (cf. p. 101).

[121] Yotam LIVNY, Feilong YAN, Matt OLSON et al. « Automatic reconstruction of tree skeletal structures from point clouds ». In : *ACM Transactions on Graphics* 29.6 (2010), p. 1 (cf. p. 83, 123).

[122] William E. LORENSEN et Harvey E. CLINE. « Marching cubes : A high resolution 3D surface construction algorithm ». In : *Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH '87* 21.4 (1987), p. 163–169 (cf. p. 49).

[123] Kl LOW. « Linear Least-squares Optimization for Point-to-plane ICP Surface Registration ». In : *Chapel Hill, University of North Carolina* February (2004), p. 2–4 (cf. p. 69).

[124] H-G H.-G. H.G. MAAS, A. BIENERT, St SCHELLER et al. « Automatic forest inventory parameter determination from terrestrial laser scanner data ». In : *International Journal of Remote Sensing* 29.5 (mar. 2008), p. 1579–1593 (cf. p. 77).

[125] Henri MAITRE. « Un panorama de la transformation de Hough ». In : *Traitement du Signal* 2.4 (1985), p. 305–317 (cf. p. 98).

[126] T. MALISIEWICZ et a. a. EFROS. « Improving Spatial Support for Objects via Multiple Segmentations ». In : *Procedings of the British Machine Vision Conference 2007*. British Machine Vision Association, 2007, p. 55.1–55.10 (cf. p. 78).

[127] Sam MARDEN et Jose GUIVANT. « Improving the Performance of ICP for Real-Time Applications using an Approximate Nearest Neighbour Search ». In : *Proceedings of Australasian Conference on Robotics and Automation*. 2012, p. 3–5 (cf. p. 58, 69).

[128] Olivier MARTIN DUCUP, Robert SCHNEIDER et Richard A. FOURNIER. « Response of sugar maple (Acer saccharum, Marsh.) tree crown structure to competition in pure versus mixed stands ». In : *Forest Ecology and Management* 374 (2016), p. 20–32 (cf. p. 152, 173, 180).

[129] RR MARTIN, Stroud IA et AD MARSHALL. *Data reduction for reverse engineering*. Rapp. tech. 1997, p. 111 (cf. p. 74).

[130] Tobias MARTIN, Juan MONTES, Jean-Charles BAZIN et al. « Topology-aware Reconstruction of Thin Tubular Structures ». In : {*SIGGRAPH*} {*Asia*} *2014* {*Technical*} {*Briefs*} (2014), 12 :1–12 :4 (cf. p. 122).

[131] M. MCDANIEL, T. NISHIHATA, C. BROOKS et al. « Terrain classification and identification of tree stems using ground based LiDAR ». In : *Journal of Field Robotics* (2012), p. 1–49 (cf. p. 124, 153, 155).

[132] Quentin MÉRIGOT, Maks OVSJANIKOV, Leonidas J. GUIBAS et al. « Voronoi-based curvature and feature estimation from point clouds ». In : *IEEE Transactions on Visualization and Computer Graphics* 17.6 (déc. 2011), p. 743–756 (cf. p. 64).

[133] Jérôme METZ, Dominik SEIDEL, Peter SCHALL et al. « Crown modeling by terrestrial laser scanning as an approach to assess the effect of aboveground intra- and interspecific competition on tree growth ». In : *Forest Ecology and Management* 310 (2013), p. 275–288 (cf. p. 152, 173, 174, 180).

[134] Mark MEYER, Mathieu DESBRUN, Peter SCHR et al. « Discrete Differential-Geometry Operators for Triangulated 2-Manifolds ». In : *Visualization and Mathematics III* (2002), p. 113–134. arXiv : arXiv : 1011 . 1669v3 (cf. p. 63).

[135] Mark MEYER, Mathieu DESBRUN, Peter SCHRÖDER et al. « Discrete differential-geometry operators for triangulated 2-manifolds ». In : *Visualization and mathematics III*. Springer, 2003, p. 35–57 (cf. p. 46).

[136] Yongwei MIAO, Renato PAJAROLA et Jieqing FENG. « Curvature-aware adaptive re-sampling for point-sampled geometry ». In : *CAD Computer Aided Design* 41.6 (2009), p. 395–403 (cf. p. 74).

[137] Niloy J. MITRA, An NGUYEN et Leonidas GUIBAS. « Estimating Surface Normals in Noisy Point Cloud Data ». In : *International Journal of Computational Geometry and Applications* 14.4-5 (2004), p. 261–276 (cf. p. 56).

[138] Carsten MOENNING et Na DODGSON. « A new point cloud simplification algorithm ». In : *Proc. Int. Conf. on Visualization, Imaging . . .* Viip (2003), p. 8–10 (cf. p. 56, 72, 75).

[139] Carsten MOENNING et Neil a DODGSON. « Intrinsic point cloud simplification ». In : *Proc. 14th GraphiCon* 14.September (2004), p. 6–10 (cf. p. 75).

[140] Jules MOREL, Alexandra BAC et Cedric VEGA. « Terrain Model Reconstruction from Terrestrial LiDAR Data Using Radial Basis Functions ». In : *IEEE Computer Graphics and Applications* (2017) (cf. p. 42).

[141] Bryan S MORSE, Terry S YOO, Penny RHEINGANS et al. « Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions ». In : *ACM SIGGRAPH 2005 Courses*. ACM. 2005, p. 78 (cf. p. 91).

[142] L. Monika MOSKAL et Guang ZHENG. « Retrieving forest inventory variables with terrestrial laser scanning (TLS) in urban heterogeneous forest ». In : *Remote Sensing* 4.1 (2012), p. 1–20 (cf. p. 153).

[143] Priyanka MUKHOPADHYAY et Bidyut B. CHAUDHURI. « A survey of Hough Transform ». In : *Pattern Recognition* 48.3 (2014), p. 993–1010 (cf. p. 98).

[144] Erik NÆSSET et Tonje ØKLAND. « Estimating tree height and tree crown properties using airborne scanning laser in a boreal nature reserve ». In : *Remote Sensing of Environment* 79.1 (2002), p. 105–115 (cf. p. 152).

[145] N E L NARVÁEZ, E A L NARVÁEZ et J W BRANCH. « Automatic construction of NURBS surfaces from unorganized points ». In : *Reconstrucción automática de superficies NURBS a partir de puntos no ordenados* 78.166 (2011), p. 133–141 (cf. p. 43, 88).

[146] Pushmeet Kohli NATHAN SILBERMAN DEREK HOIEM et Rob FERGUS. « Indoor Segmentation and Support Inference from RGBD Images ». In : *ECCV*. 2012 (cf. p. 38).

[147] NATURAL RESOURCES CANADA. *Natural Resources Canada : Remote sensing in forestry*. 2017 (cf. p. 28).

[148] NORTHEASTERN STATES RESEARCH COOPERATIVE. *Evaluation of Northern Hardwood Forest Structure Using Terrestrial Laser Scanning (Ground-Based LIDAR)*. 2008 (cf. p. 39).

[149] Y. OHTAKE, A. BELYAEV et H.P. SEIDEL. « A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions ». In : *2003 Shape Modeling International.* 2003 (2003), p. 153–161 (cf. p. 43, 88, 91, 92).

[150] Yutaka OHTAKE, Alexander BELYAEV et Hans Peter SEIDEL. « 3D scattered data approximation with adaptive compactly supported radial basis functions ». In : *Proceedings - Shape Modeling International SMI 2004*. 2004, p. 31–39 (cf. p. 92).

[151] Kenneth OLOFSSON, Johan HOLMGREN et Håkan OLSSON. « Tree stem and height measurements using terrestrial laser scanning and the RANSAC algorithm ». In : *Remote Sensing* 6.5 (2014), p. 4323–4344 (cf. p. 124, 125, 153, 173, 180).

[152] Ahlem OTHMANI, Alexandre PIBOULE, M. KREBS et al. « Towards automated and operational forest inventories with T-Lidar ». In : *SilviLaser*. 2011, p. 1–9 (cf. p. 77, 124, 153, 155).

[153] Nobuyuki OTSU. « A threshold selection method from gray-level histograms ». In : *Automatica* 11.285-296 (1975), p. 23–27 (cf. p. 131).

[154] István PÁL. « Measurements of Forest Inventory Parameters on Terrestrial Laser Scanning Data Using Digital Geometry and Topology ». In : *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XXXVII.Part B3b (2008), p. 373–380 (cf. p. 124, 153).

[155] M PAULY. « Spectral processing of point-sampled geometry ». In : *International Conference on Computer Graphics and Interactive Techniques* August (2001), p. 379–386 (cf. p. 71).

[156] M. PAULY, M. GROSS et L.P. P KOBBELT. « Efficient simplification of point-sampled surfaces ». In : *13th IEEE Visualization conference.* Section 4 (2002), p. 163–170 (cf. p. 73–75, 104, 105).

[157] M PAULY, NJ MITRA et L GUIBAS. « Uncertainty and variability in point cloud surface data ». In : *Symposium on point-based graphics* 84 (2004), p. 77–84 (cf. p. 74).

[158] Mark PAULY, Leif P KOBBELT et Markus GROSS. *Multiresolution modeling of point-sampled geometry*. Rapp. tech. 378. 2002, p. 1–9 (cf. p. 71, 73).

[159] PCL. *PCL documentation web page*. 2017 (cf. p. 38).

[160] Norbert PFEIFER, Ben GORTE et Daniel WINTERHALDER. « Automatic Reconstruction of Single Trees From Terrestrial Laser Scanner Data ». In : *Proceedings of 20th ISPRS Congress*. 2004, p. 114–119 (cf. p. 125).

[161] Norbert PFEIFER et Daniel WINTERHALDER. « Modelling of tree cross sections from terrestrial laser scanning data with free-form curves ». In : *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 36.8/W2 (2004), p. 76–81 (cf. p. 51, 124, 153, 155).

[162] Matt PHARR et Greg HUMPHREYS. *Physically Based Rendering : From Theory to Implementation*. 2004, p. 1021. arXiv : arXiv:1011.1669v3 (cf. p. 158).

[163] William H PRESS. *Numerical recipes 3rd edition : The art of scientific computing*. Cambridge university press, 2007 (cf. p. 106).

[164] Pyare PUESCHEL, Glenn NEWNHAM, Gilles ROCK et al. « The influence of scan mode and circle fitting on tree stem detection, stem diameter and volume extraction from terrestrial laser scans ». In : *ISPRS Journal of Photogrammetry and Remote Sensing* 77 (2013), p. 44–56 (cf. p. 124).

[165] Ministère des Forêts de la Faune et des Parcs du QUÉBEC. *Zones de végétation et domaines bioclimatiques du Québec*. 2017 (cf. p. 160).

[166] J. R. et Carl de BOOR. « A Practical Guide to Splines. » In : *Mathematics of Computation* 34.149 (1980), p. 325. arXiv : 1501.02634 (cf. p. 44, 45).

[167] Tahir RABBANI et Frank Van Den HEUVEL. « Efficient Hough Transform for Automatic Detection of Cylinders in Point Clouds ». In : *ISPRS Workshop on Laser Scanning* 3 (2005), p. 60–65 (cf. p. 98).

[168] Pasi RAUMONEN, E. CASELLA, Kim CALDERS et al. « Massive-Scale Tree Modelling From Tls Data ». In : *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* II-3/W4.March (2015), p. 189–196 (cf. p. 145, 154).

[169] Pasi RAUMONEN, Mikko KAASALAINEN, Markku ÅKERBLOM et al. « Fast Automatic Precision Tree Models from Terrestrial Laser Scanner Data ». In : *Remote Sensing* 5.2 (jan. 2013), p. 491–520 (cf. p. 124, 154).

[170] Joris RAVAGLIA, Alexandra BAC et Richard FOURNIER. « Segmentation de nuages de points par octrees et analyse en composantes principales ». In : *GTMG 2014*. Lyon, France, 2014, p. 2–6 (cf. p. 60).

[171] Joris RAVAGLIA, Alexandra BAC et Richard A FOURNIER. « Extraction of tubular shapes from dense point clouds and application to tree reconstruction from laser scanned data ». In : *Computers & Graphics* 66 (2017), p. 23–33 (cf. p. 43, 153, 154).

[172] Joris RAVAGLIA, Alexandra BAC et Alexandre PIBOULE. « Laser-scanned tree stem filtering for forest inventories measurements ». In : *Proceedings of the DigitalHeritage 2013 - Federating the 19th Int'l VSMM, 10th Eurographics GCH, and 2nd UNESCO Memory of the World Conferences, Plus Special Sessions fromCAA, Arqueologica 2.0 et al.* 1 (2013), p. 649–652 (cf. p. 155).

[173] Joris RAVAGLIA, Richard FOURNIER et Alexandra BAC. « Spatial vegetation density index from terrestrial laser scanner : a continuous approach applied to forestry ». In : *IGARSS 2014*. Québec, Canada, 2014 (cf. p. 58).

[174] Szymon RUSINKIEWICZ. « Estimating curvatures and their derivatives on triangle meshes ». In : *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*. IEEE. 2004, p. 486–493 (cf. p. 46).

[175] Radu Bogdan RUSU et Steve COUSINS. « 3D is here : Point Cloud Library (PCL) ». In : *Proceedings - IEEE International Conference on Robotics and Automation*. 2011. arXiv : 1409.1556 (cf. p. 65, 71).

[176] Radu Bogdan RUSU, Zoltan Csaba MARTON, Nico BLODOW et al. « Towards 3D Point cloud based object maps for household environments ». In : *Robotics and Autonomous Systems* 56.11 (nov. 2008), p. 927–941 (cf. p. 53, 62, 77).

[177] Hanan SAMET. « Implementing ray tracing with octrees and neighbor finding ». In : *Computers and Graphics* 13.4 (1989), p. 445–460 (cf. p. 60).

[178] A. D. SAPPA et M. DEVY. « Fast range image segmentation by an edge detection strategy ». In : *Proceedings of International Conference on 3-D Digital Imaging and Modeling, 3DIM*. T. 2001-Janua. 2001, p. 292–299 (cf. p. 76).

[179] A. SCHILLING et H.G. MAAS. « Automatic reconstruction of skeletal structures from TLS forest scenes ». In : *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* II-5.June (2014), p. 321–328 (cf. p. 80).

[180] Anita SCHILLING, Anja SCHMIDT et Hans-Gerd MAAS. « Automatic Tree Detection and Diameter Estimation in Terrestrial Laser Scanner Point Clouds ». In : *Proceedings of the 16th Computer Vision Winter Workshop*. 2011, p. 75–83 (cf. p. 125, 145, 153).

[181] Ruwen SCHNABEL, Roland WAHL et Reinhard KLEIN. « Efficient RANSAC for Point-Cloud Shape Detection ». In : *Computer graphics forum*. T. 26. 2. Wiley Online Library. 2007, p. 214–226 (cf. p. 51, 53, 62, 94, 97).

[182] Peter SCHRÖDER. *Course CS 117 : Discrete differential geometry*. Rapp. tech. California Institute of Technology, 2012 (cf. p. 65).

[183] Mahadev SHARMA et Richard G ODERWALD. « Dimensionally compatible volume and taper equations ». In : *Canadian Journal of Forest Research* 31.5 (2001), p. 797–803 (cf. p. 23, 26).

[184] Bao Quan SHI et Jin LIANG. « A cylindrical neighborhood for multi-view range images processing ». In : *Optics and Lasers in Engineering* 66 (2015), p. 240–248 (cf. p. 56).

[185] Bao Quan SHI, Jin LIANG et Qing LIU. « Adaptive simplification of point cloud using k-means clustering ». In : *CAD Computer Aided Design* 43.8 (2011), p. 910–922 (cf. p. 74).

[186] Merlijn SIMONSE, Tobias ASCHOFF, Heinrich SPIECKER et al. « Automatic determination of forest inventory parameters using terrestrial laser scanning ». In : *Proceedings of the ScandLaser Scientific Workshop on Airborne Laser Scanning of Forests*. T. 2003. 2003, p. 252–258 (cf. p. 98, 125, 153).

[187] Hao SONG et Hsi Yung FENG. « A global clustering approach to point cloud simplification with a specified data reduction ratio ». In : *CAD Computer Aided Design* 40.3 (2008), p. 281–292 (cf. p. 73).

[188] Hao SONG et Hsi Yung FENG. « A progressive point cloud simplification algorithm with preserved sharp edge data ». In : *International Journal of Advanced Manufacturing Technology* 45.5-6 (2009), p. 583–592 (cf. p. 73).

[189] S SOTOODEH. « Outlier Detection in Laser Scanner Point Clouds ». In : *Iaprs* Volume XXX.Part 5 (2006), p. 297–302 (cf. p. 71).

[190] Noriaki SUETAKE, Eiji UCHINO et Kanae HIRATA. « Generalized fuzzy Hough transform for detecting arbitrary shapes in a vague and noisy image ». In : *Soft Computing* 10.12 (2006), p. 1161–1168 (cf. p. 97).

[191] Andrea TAGLIASACCHI, Hao ZHANG et Daniel COHEN-OR. « Curve skeleton extraction from incomplete point cloud ». In : *ACM SIGGRAPH 2009 papers on - SIGGRAPH '09* 28.3 (2009), p. 1 (cf. p. 62, 81, 82, 123).

[192] P. W. THEILER, J. D. WEGNER et K. SCHINDLER. « Markerless point cloud registration with keypoint-based 4-points congruent sets ». In : *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* II-5/W2.November (2013), p. 283–288 (cf. p. 74).

[193] Brian TOBIN, Kevin BLACK, Bruce OSBORNE et al. « Assessment of allometric algorithms for estimating leaf biomass, leaf area index and litter fall in different-aged Sitka spruce forests ». In : *Forestry* 79.4 (2006), p. 453–465 (cf. p. 151).

[194] Phil TORR et Andrew ZISSERMAN. « Robust computation and parametrization of multiple view relations ». In : *Proceedings of the {IEEE} International Conference on Computer Vision* (1998), p. 727–732 (cf. p. 96).

[195] Greg TURK. « Re-tiling polygonal surfaces ». In : *ACM SIGGRAPH Computer Graphics* 26.July 1992 (1992), p. 55–64 (cf. p. 75).

[196] Greg TURK et James F O'BRIEN. « Shape transformation using variational implicit functions ». In : *Proceedings of the 26th annual conference on Computer graphics and interactive techniques SIGGRAPH 99* 33.Annual Conference Series (1999), p. 335–342 (cf. p. 48, 90).

[197] Greg TURK et James F O'BRIEN. « Modelling with implicit surfaces that interpolate ». In : *ACM Transactions on Graphics* 21.4 (2002), p. 855–873 (cf. p. 48).

[198] Marek VANČO, Bernd HAMANN et Guido BRUNNETT. « Surface reconstruction from unorganized point data with quadrics ». In : *Computer Graphics Forum* 27.6 (2008), p. 1593–1606 (cf. p. 43, 53, 77, 88).

[199] Kenneth VERSPRILLE. « Computer-Aided Design Applications of the Rational B-Spline Approximation Form ». Thèse de doct. 1975 (cf. p. 44).

[200] George VOSSELMAN et Hans-Gerd MAAS. *Airborne and Terrestrial Laser Scanning*. 2010 (cf. p. 152).

[201] Miao WANG, Yi-hsing TSENG, Laser SCANNING et al. « Lidar data segmentation and classification based on octree structure ». In : *International Society for Photogrammetry and Remote Sensing* 2.1 (2004), p. 1–6 (cf. p. 60).

[202] Yutao WANG et Hsi Yung FENG. « Outlier detection for scanned point clouds using majority voting ». In : *CAD Computer Aided Design* 62 (2015), p. 31–43 (cf. p. 71).

[203] P WEZYK, K KOZIOŁ, M GLISTA et al. *Terrestrial laser scanning versus traditional forest inventory first results from the polish forests*. 2007 (cf. p. 153).

[204] Donna J. WILLIAMS et Mubarak SHAH. « A Fast algorithm for active contours and curvature estimation ». In : *CVGIP : Image Understanding* 55.1 (1992), p. 14–26 (cf. p. 126).

[205] Michael A. WULDER, Joanne C. WHITE, Richard A. FOURNIER et al. « Spatially Explicit Large Area Biomass Estimation : Three Approaches Using Forest Inventory and Remotely Sensed Imagery in a GIS ». In : *Sensors* 8.1 (2008), p. 529–560 (cf. p. 151).

[206] Zhouxin XI, Christopher HOPKINSON et Laura CHASMER. « Automating Plot-Level Stem Analysis from Terrestrial Laser Scanning ». In : *Forests* 7.12 (2016), p. 252 (cf. p. 153).

[207] Chenyang XU et Jerry L. PRINCE. « Snakes, shapes, and gradient vector flow ». In : *IEEE Transactions on Image Processing* 7.3 (1998), p. 359–369 (cf. p. 126).

[208] Hui XU, Nathan GOSSETT et Baoquan CHEN. « Knowledge and heuristic-based modeling of laser-scanned trees ». In : *ACM Transactions on Graphics* 26.4 (oct. 2007), 19–es (cf. p. 56, 83, 84, 123).

[209] DM YAN, Julien WINTZ, Bernard MOURRAIN et al. « Efficient and robust tree model reconstruction from laser scanned data points ». In : *Proceedings of the 11th ...* 2009, p. 572–576 (cf. p. 153).

[210] Jian YAO, Mauro R. RUGGERI, Pierluigi TADDEI et al. « Automatic scan registration using 3D linear and planar features ». In : *3D Research* 1.3 (2010), p. 1–18 (cf. p. 69).

[211] Mustafa YILMAZ et Murat UYSAL. « A COMPARATIVE STUDY OF CURVATURE AND GRID DATA REDUCTION ALGORITHMS FOR LIDAR-DERIVED DIGITAL TERRAIN MODELS Mustafa ». In : *Proceedings, 6th International Conference on Cartography and GIS*. June. Albena, Bulagria, 2016, p. 13–17 (cf. p. 74).

[212] Zhiwen YU et Hau San WONG. « An efficient local clustering approach for simplification of 3D point-based computer graphics models ». In : *2006 IEEE International Conference on Multimedia and Expo, ICME 2006 - Proceedings*. T. 2006. 2006, p. 2065–2068 (cf. p. 74).

[213] Zhiwen YU, Hau San WONG, Hong PENG et al. « ASM : An adaptive simplification method for 3D point-based models ». In : *CAD Computer Aided Design* 42.7 (2010), p. 598–612 (cf. p. 74).

[214] Faisal ZAMAN, Ya Ping WONG et Boon Yian NG. « Density-based denoising of point cloud ». In : *Lecture Notes in Electrical Engineering*. T. 398. 2017, p. 287–295. arXiv : 1602.05312 (cf. p. 71).

[215] Mengmi ZHANG. « Accurate Sphere Marker-Based Registration System of 3D Point Cloud Data in Applications of Shipbuilding Blocks ». In : *Journal of Industrial and Intelligent Information* 3.4 (2015), p. 318–323 (cf. p. 97).

[216] Zhengyou ZHANG. « Iterative point matching for registration of free-form curves and surfaces ». In : *International Journal of Computer Vision* 13.2 (1994), p. 119–152 (cf. p. 69).

[217] P ZHAO, Y WANG et Q HU. « A FEATURE PRESERVING ALGORITHM FOR POINT CLOUD SIMPLIFICATION BASED ON HIERARCHICAL CLUSTERING ». In : (2016), p. 5581–5584 (cf. p. 64, 74).

[218] Guiyun ZHOU, Bin WANG et Ji ZHOU. « Automatic registration of tree point clouds from terrestrial LiDAR scanning for reconstructing the ground scene of vegetated surfaces ». In : *IEEE Geoscience and Remote Sensing Letters* 11.9 (2014), p. 1654–1658 (cf. p. 69).